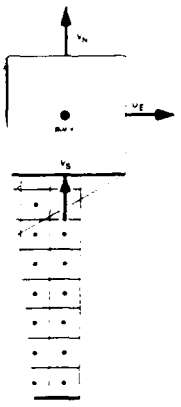
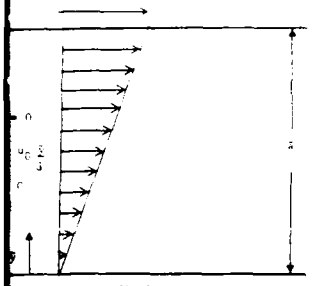
TECHNICAL REPORT HL-89-8

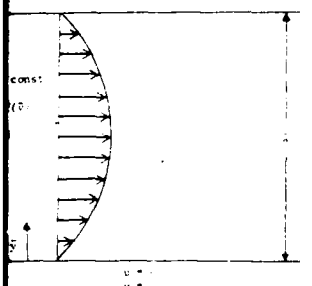
(2)

US Army Corps of Engineers





Case without a pressure gradient



Case with a pressure gradient

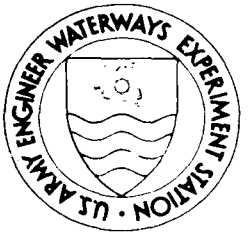
DEVELOPMENT OF AN EFFICIENT SOLUTION SCHEME FOR INCOMPRESSIBLE STEADY-STATE FLOW

by

Jeffery P. Holland

Hydraulics Laboratory

DEPARTMENT OF THE ARMY
Waterways Experiment Station, Corps of Engineers
PO Box 631, Vicksburg, Mississippi 39181-0631



April 1989
Final Report

HYDRAULICS LABORATORY

Prepared for DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) Technical Report HL-89-8	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION USAEWES Hydraulics Laboratory 6b OFFICE SYMBOL (If applicable) CEWES-HS-R	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and ZIP Code) PO Box 631 Vicksburg, MS 39181-0631	7b ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION US Army Corps of Engineers 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code) Washington, DC 20314-1000	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO. See Reverse

11 TITLE (Include Security Classification)
Development of an Efficient Solution Scheme for Incompressible Steady-State Flow

12 PERSONAL AUTHOR(S)
Holland, Jeffery P.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final Report	FROM _____ TO _____	April 1989	193

16 SUPPLEMENTARY NOTATION Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161. Originally submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Civil Engineering to (Continued)

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Convergence acceleration
Finite-volume scheme
Incompressible flow
Multigrid
Navier-Stokes equations
Predictor-corrector scheme
Pseudocompressibility

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This study advances the understanding of the numerical simulation of steady-state, incompressible flows through development of a new solution methodology for the two-dimensional Navier-Stokes equations. The study presents details of a numerical scheme formulated specifically to simulate flows generally observed in the approaches to hydraulic structures. An explicit predictor-corrector finite volume relaxation scheme is coupled with a pseudocompressibility methodology to integrate the governing equations of motion and continuity. The use of the pseudocompressibility concept negates the need for the solution of a Poisson equation relating the pressure and flux fields. Results from the simulation of four model case studies show the efficacy of the relaxation scheme.

To accelerate the convergence of the basic relaxation scheme, a multigrid algorithm is coupled with the predictor-corrector. The multigrid approach is patterned after the works of Brandt and Jameson and Yoon. Results from additional simulation of the four (continued)

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	
22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code) / 22c OFFICE SYMBOL

10. WORK UNIT ACCESSION NO. (Continued).

Funding provided by Civil Works Investigation Work Unit No. 32319, sponsored by Headquarters, US Army Corps of Engineers.

16. SUPPLEMENTARY NOTATION (Continued).

Colorado State University, Fort Collins, CO

19. ABSTRACT (Continued).

model case studies conclusively show the validity and attractiveness of employing the multigrid approach in the simulation of incompressible, steady-state flows. The flow fields numerically generated through inclusion of the multigrid algorithm are just as accurate as those computed with the basic relaxation scheme alone. In addition, the model test case results obtained with the multigrid algorithm are generally from 3 to 12 times more efficient in reaching a predefined convergence tolerance than their relaxation scheme-only counterparts based on computer resource usage. The inclusion of the multigrid algorithm requires more computational overhead than the basic relaxation scheme computations, however, producing up to a nine-fold increase in the time per iteration compared to these latter computations. This computational overhead is easily compensated for by the reduction in the number of iterations required to reach convergence with the multigrid algorithm included.

The optimal multigrid setup was that which utilizes the maximum number of total grids allowable given the resolution on the finest grid. In addition, convergence is most faithfully accelerated by the use of one or two relaxation sweeps on the finest grid in concert with the use of one relaxation sweep on each of the coarser grids employed.

PREFACE

The study described herein was performed at the US Army Engineer Waterways Experiment Station (WES) during the period 1985-1988 for the Headquarters, US Army Corps of Engineers (USACE), as part of the Civil Works Research and Development Program. Funds were allocated under the Repair, Evaluation, Maintenance, and Rehabilitation Program, Civil Works Investigation Work Unit No. 32319, "Predictive Techniques for Approach Flow Conditions to Spillways and Other Structures," under USACE Program Monitor Mr. Glenn Drummond.

This study was accomplished under the direction of Messrs. F. A. Herrmann, Jr., Chief of the Hydraulics Laboratory, WES; J. L. Grace, Jr., former Chief of the Hydraulic Structures Division; and G. A. Pickering, Chief of the Hydraulic Structures Division. This report was written by Dr. J. P. Holland, Chief of the Reservoir Water Quality Branch.

This report was also submitted to the Academic Faculty of Colorado State University, Fort Collins, CO, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Civil Engineering.

COL Dwayne G. Lee, EN, is the Commander and Director of WES. Dr. Robert W. Whalin is the Technical Director.

TABLE OF CONTENTS

		Page
PREFACE		i
LIST OF TABLES		v
LIST OF FIGURES		vii
CHAPTER 1:	INTRODUCTION	1
CHAPTER 2:	RESEARCH OBJECTIVE AND JUSTIFICATION	5
CHAPTER 3:	REVIEW OF NUMERICAL INCOMPRESSIBLE FLOW SOLUTIONS	7
3.1	Overview	7
3.2	Finite Difference Solution vs. Finite Element Solution	8
3.3	Explicit vs. Implicit Methods	9
3.4	Staggered vs. Regular Grids	14
3.5	Primitive Variable Solution vs. Other Formulations	15
3.6	Solution Scheme Accuracy/Mass Conservation	17
3.7	Poisson vs. Chorin Pressure Solution	18
3.8	Convergence Acceleration	21
3.9	Summary of Methodology Chosen for Further Evaluation	24
CHAPTER 4:	GOVERNING EQUATIONS FOR INCOMPRESSIBLE FLOW	25
4.1	Overview	25
4.2	Conservation Equations	25
4.3	Pressure Solution Equations	30

TABLE OF CONTENTS (Continued)

Page

4.4 Convergence Acceleration Methodology..... 36

CHAPTER 5: NUMERICAL SOLUTION METHODOLOGY..... 43

5.1 Introduction..... 43

5.2 Problem Discretization..... 43

5.3 Primary Solution Steps..... 49

5.4 Incorporation of Multigrid Methodology..... 54

CHAPTER 6: ADDITIONAL NUMERICAL CRITERIA..... 68

6.1 Introduction..... 68

6.2 Boundary Condition Implementation..... 68

6.3 Initial Condition Specification..... 73

6.4 Stability Criteria..... 73

6.5 Maximum Number of Grids Allowable..... 75

CHAPTER 7: MASS CONSERVATION CASE STUDY..... 77

7.1 Background..... 77

7.2 Case Study Details..... 79

7.3 Results of Potential Flow Simulations..... 82

7.4 Initial Trends for Further Evaluation..... 96

CHAPTER 8: VISCOUS FLOW RESULTS..... 98

8.1 Case Study Details..... 98

8.2 Results of Couette Flow Simulation..... 101

8.3 Results of Stationary Viscous Channel Flow..... 123

CHAPTER 9: CONCLUSIONS AND RECOMMENDATIONS..... 133

9.1 Chapter Summary..... 133

9.2 Discussion and Conclusions..... 133

9.3 Recommendations for Future Work..... 139

TABLE OF CONTENTS (Concluded)

Page

REFERENCES 142

APPENDIX A: DESCRIPTION OF COMPUTATIONAL SOFTWARE 148

A.1 Introduction 148

A.2 Basic Computer Formulation 148

A.3 Synopsis of Subroutine Components 150

A.4 Input Specification 157

A.5 Output Specification 163

APPENDIX B: DETAILS OF VISCOUS COMPUTATIONS 165

B.1 Introduction 165

B.2 Generalized Laplacian Formulation 165

B.3 Calculation and Restriction of No-Slip Contribution 169

B.4 Velocity Boundary Condition for No-Slip Boundaries 170

APPENDIX C: PRESSURE PROLONGATION PROCEDURE 173

C.1 Introduction 173

C.2 Bilinear Interpolation Scheme 173

C.3 Location of Finer-Grid Points 176

APPENDIX D: NOTATION 177

LIST OF TABLES

Table		Page
7.1	Summary of Convergence Norms for Multiple Grid Runs: Potential Flow, 21-by-21 Orthogonal Grid Problem, Convergence Tolerance - 0.0001.....	83
7.2	Comparison of Convergence Properties for Multiple Grid Runs: Potential Flow, 21-by-21 Orthogonal Grid, Convergence Tolerance - 0.0001.....	89
7.3	Summary of Convergence Norms for Multiple Grid Runs: Potential Flow, 21-by-21 Nonorthogonal Grid, Convergence Tolerance - 0.001.....	91
7.4	Comparison of Convergence Properties for Multiple Grid Runs: Potential Flow, 21-by-21 Nonorthogonal Grid, Convergence Tolerance - 0.001.....	91
8.1	Summary of Convergence Norms for Multiple Grid Grid Runs: Couette Flow, 21-by-21 Grid Problem, Convergence Tolerance - 0.0001, Reynolds Number = 100.....	102
8.2	Comparison of Convergence Properties for Multiple Grid Runs: Couette Flow, 21-by-21 Grid Problem, Convergence Tolerance - 0.0001, Reynolds Number = 100.....	107
8.3	Summary of Convergence Norms for Multiple Grid Runs: Couette Flow, 21-by-21 Grid Problem, Convergence Tolerance - 0.0001, Reynolds Number = 400.....	108
8.4	Comparison of Convergence Properties for Multiple Grid Runs: Couette Flow, 21-by-21 Grid, Problem, Convergence Tolerance - 0.0001, Reynolds Number = 400.....	112
8.5	Summary of Convergence Norms for Multiple Grid Runs: Couette Flow, 41-by-41 Grid Problem, Convergence Tolerance - 0.0001, Reynolds Number = 100.....	121
8.6	Comparison of Convergence Properties for Multiple Grid Runs: Couette Flow, 41-by-41 Grid Problem, Convergence Tolerance - 0.0001, Reynolds Number = 100.....	122
8.7	Summary of Convergence Norms for Multiple Grid Runs: Stationary, Viscous Flow, 21-by-21 Grid Problem, Convergence Tolerance - 0.0001, Reynolds Number = 100.....	124

LIST OF TABLES (Concluded)

Table		Page
8.8	Comparison of Convergence Properties for Multiple Grid Runs: Stationary, Viscous Flow, 21-by-21 Grid Problem Convergence Tolerance - 0.0001, Reynolds Number = 100.....	126
8.9	Summary of Convergence Norms for Multiple Grid Runs: Stationary, Viscous Flow, 41-by-41 Grid Problem, Convergence Tolerance - 0.0001, Reynolds Number = 100.....	128
8.10	Comparison of Convergence Properties for Multiple Grid Runs: Stationary, Viscous Flow, 41-by-41 Grid Problem, Convergence Tolerance - 0.0001, Reynolds Number = 100.....	128

LIST OF FIGURES

Figure		Page
5.1	Arbitrary physical domain	44
5.2	Computational domain with variable definition	45
5.3	Restriction methodology for fluxes	55
5.4	Relationships for pressure prolongation	66
6.1	Labeling system for cell-centered quantities in neighboring grid cells	69
6.2	Schematic of a given cell (cc) whose east face is coincident with a physical boundary	70
7.1	21-by-21 orthogonal and nonorthogonal grids	81
7.2	Schematic of mass conservation test setup	82
7.3	Convergence history for divergence of velocity, 21-by-21 orthogonal grid, potential flow, multiple grid runs	84
7.4	Convergence history for U flux, 21-by-21 orthogonal grid, potential flow, multiple grid runs	85
7.5	Convergence history for V flux, 21-by-21 orthogonal grid, potential flow, multiple grid runs	86
7.6	Convergence history for divergence of velocity, 21-by-21 nonorthogonal grid, potential flow, multiple grid runs	92
7.7	Convergence history for U flux, 21-by-21 nonorthogonal grid, potential flow, multiple grids runs	93
7.8	Convergence history for V flux, 21-by-21 nonorthogonal grid, potential flow, multiple grid runs	94

LIST OF FIGURES (Continued)

Figure		Page
8.1	41-by-41 grid for Couette flow simulation	98
8.2	Schematics for Couette flow test cases	100
8.3	Convergence history for divergence of velocity 21-by-21 grid, Couette flow, Re = 100, multiple grid runs	103
8.4	Convergence history for U flux, 21-by-21 grid, Couette flow, Re = 100, multiple grid runs	104
8.5	Convergence history for V flux, 21-by-21 grid, Couette flow, Re = 100, multiple grid runs	105
8.6	Convergence history for divergence of velocity, 21-by-21 grid, Couette flow, Re = 400, multiple grid runs	109
8.7	Convergence history for U flux, 21-by-21 grid, Couette flow, Re = 400, multiple grid runs	110
8.8	Convergence history for V flux, 21-by-21 grid, Couette flow, Re = 400, multiple grid runs	111
8.9	Convergence history for divergence of velocity, 41-by-41 grid, Couette flow, Re = 100, multiple grid runs	115
8.10	Convergence history for U flux, 41-by-41 grid, Couette flow, Re = 100, multiple grid runs	117
8.11	Convergence history for V flux, 41-by-41 grid, Couette flow, Re = 100, multiple grid runs	119
8.12	Convergence history for divergence of velocity, 21-by-21 grid, stationary, viscous flow, Re = 100 multiple grid runs	125
8.13	Convergence history for U flux, 21-by-21 grid, stationary, viscous flow, Re = 100, multiple grid runs	125
8.14	Convergence history for V flux, 21-by-21 grid, stationary, viscous flow, Re = 100, multiple grid runs	126

LIST OF FIGURES (Concluded)

Figure		Page
8.15	Convergence history for divergence of velocity, 41-by-41 grid, stationary, viscous flow, Re = 100, multiple grid runs	129
8.16	Convergence history for U flux, 41-by-41 grid, stationary, viscous flow, Re = 100, multiple grid runs	130
8.17	Convergence history for V flux, 41-by-41 grid, stationary, viscous flow, Re = 100, multiple grid runs	131
C.1	Fine-to-coarse transfer during prolongation	174

CHAPTER 1

INTRODUCTION

During the design, maintenance, and operation of hydraulic structures, large amounts of financial and temporal resource are expended. An integral part of this process involves physical and numerical simulation of the hydrodynamic characteristics of alternative structural designs and operations. Such simulations are attractive because they result in improvements to the design and operation/ maintenance aspects of projects without the costs that are associated with field experimentation. For example, prototype construction of an approach for a typical spillway at a Corps of Engineers flood control reservoir might cost $1 million (Oswalt, 1988); furthermore, this cost reflects only construction and does not include initial design and engineering costs. Should the performance of a structural design prove inadequate, additional costs on the order of the initial construction costs would be incurred. A physical model study of multiple wall designs for numerous project operations and hydrologic events would generally cost approximately $150 thousand at the Corps' Waterways Experiment Station.

Often the hydrodynamic flow fields produced by water and structure interaction are complex. An example of such complexity is the flow field produced by interaction between the approach, sump, and intake geometries of a pump station and its inflow. Irregular approach geometries often produce vorticity that can be advected into the pump

sump approach. This vorticity, if not mitigated, may result in severe damage to hydromachinery. As a consequence, this machinery may have to be repaired, or even replaced, at a substantial cost. As an example of these potential costs, the recent repair of the pumps at the Huxtable Pumping Station was completed at a cost of $400,000. Thus, these structures should be designed and operated to minimize damage to the pumps and other project components.

Traditionally, the primary simulation tool available for investigation of complex flow regimes has been physical modeling. Employing general similitude criteria, scale models are used to evaluate the efficacy of alternative designs and operations of structures for proper hydraulic performance. These models are, however, quite expensive and the expertise to build such models of sufficient detail seems to be declining. The advent of reliable two-dimensional numerical hydrodynamic models that solve the Navier-Stokes or Euler equations has assuaged this problem somewhat in that the numerical models can be used to screen potential design or operational schemes prior to physical modeling.

Unfortunately, the need to design and operate hydraulic structures cost-effectively necessitates the evaluation of three-dimensional fluid/structure interaction. Consequently, the level of detail required in these investigations demands the construction of evermore elaborate and expensive physical models. Thus, the absence of three-dimensional numerical tools to screen alternative designs places significant financial burdens on investigations of this type. Obviously, three-dimensional numerical hydrodynamic models capable of modeling near-field fluid/structure interactions are needed. It is imperative, though, that

these numerical tools be both accurate and efficient if they are to find widespread application. Sadly, many hydrodynamic solution techniques fail to satisfy either one or both of these constraints.

Accurate incompressible solution of the Navier-Stokes or Euler equations requires mass conservation. Past research (i.e., Bernard and Thompson (1984)) highlighted the importance of achieving mass conservation in such calculations. Achieving mass conservation is more difficult for incompressible flow computations than for compressible flow calculations because the incompressible continuity equation has neither a time derivative nor an advected gradient of density. Further, there is no state equation relating pressure and density for incompressible fluids. In addition, if the time derivative of density were to be retained, the maximum allowable time step for the stable solution of an incompressible flow field would be impractically small.

Errors in mass conservation for typical calculations of approach flow fields to hydraulic structures (denoted by the divergence of the velocity field) must be less than one percent in order to achieve accurate incompressible flow simulations. Indicators such as the time derivative of each of the velocities (from the momentum equations) should also have maxima on this order. Further, the solution technique chosen for a multi-dimensional hydrodynamic model should also be efficient. This is imperative since the grid systems resolved by such a code could typically have 50 to 100 thousand node points for a three-dimensional problem.

The utility of any three-dimensional numerical hydrodynamic model is based on its ability to consider the following: (a) the model must compute non-hydrostatic pressure fields; (b) the model must handle

non-uniform physical boundaries appropriately; (c) the model must consider multiple initial and boundary conditions; (d) the model must conserve mass both locally and globally and converge as quickly as possible without sacrificing accuracy; (e) the code should simulate three-dimensional, non-hydrostatic, incompressible flow in general curvilinear coordinates for both steady and unsteady situations; and (f) the code should also solve flow problems for both inviscid and viscous flows. A primary use of the code could be the simulation of steady flow phenomena in support of physical modeling activities related to hydraulic structures at the U.S. Army Waterways Experiment Station.

CHAPTER 2

RESEARCH OBJECTIVE AND JUSTIFICATION

The primary purpose of this dissertation is to advance a numerical methodology which would stand as the basis for three-dimensional (3D) numerical hydrodynamic model development. The research does not attempt to actually develop a three-dimensional code. Rather, state-of-the-art two-dimensional (2D) numerical approaches are evaluated based on model test case results, and the combination of such approaches which appears to hold the most promise for three-dimensional development is recommended.

The complexity of three-dimensional flow fields is generally such that rigorous evaluation of any three-dimensional numerical solution technique is difficult. Often the test cases used for 3D model verification are nothing more than axisymmetric 3D or two-dimensional flow fields. The approach followed in the development reported herein uses simulated results of 2D flow fields of known solution as a basis for recommending further 3D development. Such an approach is deemed equally or more rigorous than actual 3D simulation since it allows intricate evaluation and verification of model components for established problems. Further, this approach expands the bulk of 2D modeling knowledge by examining the efficacy of certain existing 2D formulations and the creation of new ones (as needed).

Another focus of this research is the development of inviscid and laminar viscous flow simulation tools. Although the flow fields near

hydraulic structures are highly turbulent, incorporation of a turbulence closure scheme is beyond the scope of the present research.

Model test cases were chosen based on the existence of their known solution, the ease with which their major flow features can be presented visually and analyzed, their acceptance as cases that numerical codes must be able to simulate accurately, and the rigor required by an algorithm for their accurate and efficient simulation. All the problems chosen are steady-state scenarios. Both accuracy and efficiency (rapid convergence) are stressed; however, when the two are in conflict, model accuracy receives priority.

The results presented herein provide insight into the difficulties associated with appropriately modeling hydrodynamic flow phenomena via numerical methods. The results illustrate an algorithm which can be used with confidence to simulate two-dimensional hydrodynamic flow fields accurately (through use of a newly developed finite volume staggered grid solution scheme) and efficiently (through the use of a multigrid method developed for use with the staggered grid solution scheme).

This dissertation describes the first use of the staggered grid scheme presented herein with MacCormack's (MacCormack, 1969) explicit finite difference solution scheme, Chorin's (Chorin, 1967) pressure solution method, and a multigrid methodology developed specifically for coupling with the other components. The insights obtained from the multigrid investigation and development may be the most important of the contributions made. Recommendations are presented for a direction for future three-dimensional numerical model development. The recommendations of this study will help guide 3D hydrodynamic model development by the Hydraulics Laboratory of the U.S. Army Corps of Engineers Waterways Experiment Station.

CHAPTER 3

REVIEW OF NUMERICAL INCOMPRESSIBLE FLOW SOLUTION

3.1 OVERVIEW

This chapter provides an analysis of some aspects of the numerical solution of the incompressible Navier-Stokes equations. Emphasis is given to those methods which appear to be eligible for use in the solution of hydrodynamic flow problems. Thus, both incompressible and compressible flow solution schemes are discussed if the given scheme appears to have potential for application, either by extension or modification, to incompressible flows.

Discussion centers around only those points for which major differences in modeling philosophy or new and innovative methodologies exist in the numerical simulation of incompressible flow. The seven points covered in depth include:

1. Finite difference solution vs. finite element solution,
2. Explicit vs. implicit methods,
3. Staggered vs. regular grid schemes,
4. Primitive variable solution vs. other formulations,
5. Solution scheme accuracy/mass conservation,
6. Poisson vs. Chorin pressure solution, and
7. Convergence acceleration.

Turbulence is not included because turbulence modeling is a topic separate from the development and implementation of Navier-Stokes solvers. In practice, it is usually necessary to use an empirical

turbulence model to account for the effects of small, sub-grid scale eddies upon the large-scale flow. This should be done, however, only after the numerical scheme has been verified for laminar flow applications. As Chen (1986) has stated, the

> "accuracy of the turbulent flow prediction depends largely on the quality of the turbulence model... . Accuracy... can be evaluated more faithfully using laminar flow problems as numerical testing cases."

A number of additional topics germain to numerical solution of the incompressible Navier-Stokes equations are omitted. An excellent review of grid generation is provided by Thompson (1984). Further, Anderson et al. (1984), Roache (1972), and Peyret and Taylor (1983) provide very complete references for the general area of computational fluid dynamics and numerical methods. Peyret and Taylor, as well as Lustman (1984), discuss spectral methods. There are also a number of excellent overviews of incompressible flow simulation including: Ferziger (1987); Aref (1986); and Orszag and Israeli (1974). Johnson (1981) provides a thorough review of the solution of the incompressible Navier-Stokes equations for reservoir hydrodynamic modeling.

The next several sections of this chapter will present an evaluation of each of the seven points listed above regarding incompressible Navier-Stokes simulation. Following these sections, the points made in each section will be summarized and a collective numerical methodology identified for further study.

3.2 FINITE DIFFERENCE SOLUTION VS. FINITE ELEMENT SOLUTION

Numerical solution of fluid flow problems was first accomplished via finite difference methods. Recently, however, finite element methods have been proposed as an alternate or even preferred method of solution (Gresho et al. (1981); Gunzburger et al. (1983); Bercovier et

al (1981); Laval (1981) are examples). Baker (1983) gives a complete overview of the method and its use in fluid problems. Additionally, Chen (1982) and Yeh (1981) propose methodologies that appear to be hybrid derivatives having properties of both finite difference and finite element schemes.

Vinokur (1976) states that finite element methods are superior to finite difference methods when complex boundaries are involved in the simulation; he observes that finite difference methods are superior to finite elements whenever complex equations are to be solved, such as the Navier-Stokes equations. Further, Vinokur claims that finite volume methods have advantages of both finite differences and finite elements. Finite difference methods have often suffered a loss of accuracy when irregular boundaries are discretized on rectangular grids. However, Thompson (1984) notes that, with the advent of boundary-fitted grid generation, finite difference methods have become practical even for irregular boundaries. Further, Peyret and Taylor (1983) show that certain finite element formulations may be equivalent to finite difference schemes for relatively simple problems. Thus, there seems to be no general reason for recommending finite difference or finite element methods a priori.

3.3 EXPLICIT VS. IMPLICIT METHODS

Explicit methods are those in which a single flow variable, pressure for example, is calculated at a new iteration or time step using only information computed at the previous iterate or time step. These methods are usually sequential in nature, allowing natural sweeping activities within the flow field to take place. These methods also often vectorize very naturally on supercomputer vector hardware.

Implicit methods, on the other hand, solve for a flow variable as a function of new and old flow information computed at the current and previous time steps or iterations. These methods generally require extensive matrix manipulation and inversion, but they allow the use of larger time steps (in theory, infinitely larger; in practice, often 2 to 10 times larger) than do explicit methods. Thus, implicit methods may be used to obtain steady-state solutions in fewer iterations than explicit methods. However, implicit methods do require more arithmetic operations for each iteration than explicit methods.

Review of both compressible and incompressible flow simulation literature shows that alternating direction-implicit (ADI) methods are widely used for numerical solutions of the Navier-Stokes equations. These schemes are attractive because they produce banded matrices that are invertible by well established sparse matrix techniques. Kwak et al. (1984) employ the Beam-Warming ADI algorithm in their development of the INS3D three-dimensional incompressible flow solver for simulation of fuel flow in the Main Space Shuttle Engines. Chang et al. (1985) give details of this simulation with the INS3D code, and Rogers et al. (1986) add additional details of simulations of flow around multiple posts with this code. The Beam-Warming scheme (see Beam and Warming (1976)) is an implicit finite difference scheme that is second-order accurate in time and space and employs approximate factorization (or flux-splitting). Approximate factorization is a process whereby the formidable matrix inversion process which would be involved in a non-iterative solution of the Navier-Stokes equations is reduced to a series of tridiagonal matrix inversion problems for which efficient solutions algorithms exist. Briley and McDonald (1973) independently developed an algorithm similar

to that of Beam and Warming. Several researchers now use the Beam-Warming scheme including Kirtley et al. (1986) for hydroturbine flow calculations and Choi and Merkle (1984) for a cascade geometry. Kim and Moin (1984) also use a version of the scheme, coupling it with a fractional-step method.

Although the Beam-Warming scheme has produced accurate results for incompressible flow, approximate factorization is not without its problems. Choi and Merkle (1984) note that the approximate factorization methodology produces a contamination of the governing equations that may dominate the other "real" terms of the equations for large time steps and greatly slow convergence to the steady state. While the authors present a pre-conditioning method which removes this concern, it is presented for two-dimensional flow only. It is unknown how cumbersome such pre-conditioning would be for the full Navier-Stokes equations. Kwak et al. (1984) present an alternate methodology for dissipating the contamination brought about, in part, by the approximate factorization which uses second and fourth-order smoothing terms to the factored Navier-Stokes equations. Although some information is presented by Chang et al. (1985) on the appropriate selection of coefficients governing the strength of these smoothing terms, their selection may still involve trial and error. And, while the factorization contamination can be minimized through appropriate selection of the time step (Steger (1978)), this action may result in the use of small time steps that negates the economy of using an implicit scheme.

In addition to approximate factorization ADI methods, there are other implicit methods that receive frequent mention in the literature.

Mastin and Thompson (1978) discuss the use of point successive over-relaxation (SOR) for implicit solution of the three-dimensional, incompressible Navier-Stokes equations. These researchers use both a one-step and a two-step method for computing new iterates of the primitive variables. The two-step method resembles an implicit predictor-corrector method. Patel and Thompson (1984) use an implicit point-SOR scheme known as red-black or checkerboard SOR to increase the vectorization potential of the computer code without the use of explicit vector coding. The use of point-SOR is contrasted with the use of the Gauss-Seidel method, both with and without relaxation added, for implicit solutions. Several researchers including Chien and Schetz (1975), Moitra (1982), Fuchs and Zhao (1984), Chen (1986), and Vanka and Misegades (1987) report the use of a Gauss-Seidel technique. In general, these researchers show that the method works well for their given problems. However, simple Gauss-Seidel will not vectorize without the use of computer-specific coding, a point which reduces the portability of such a code. As Vanka and Misegades (1987) show, however, tailoring a particular code for a given machine can result in an implicit Navier-Stokes solver which vectorizes very nicely on present supercomputers.

In sharp contrast to the available information on implicit Navier-Stokes solvers, the amount of literature about explicit methods is small. Until the advent of vectorization capabilities on supercomputers, steady-state flow solutions were much more economical with implicit methods than with explicit ones. However, given the very natural way explicit schemes vectorize, their use has increased. If care is taken, many explicit formulations will vectorize for numerous

computing architectures, thereby increasing the portability of such a code. Vanka and Misegades (1987) note that most of the first codes to take advantage of the vectorization powers of supercomputers were explicit. The works of Shang et al. (1980), Redhed et al. (1979), Chima and Johnson (1983), and Smith and Pitts (1979) are examples of such.

Although there are a multitude of explicit finite difference methodologies available, many of the explicit Navier-Stokes solutions seem to utilize the predictor-corrector scheme of MacCormack (1969). Bernard (1986) uses the MacCormack explicit scheme for solution of the two-dimensional Navier-Stokes equations and recommends its use in three dimensions. Further, MacCormack (1985) recommends his own scheme as the basic foundation of Navier-Stokes solvers for the future. This is in contrast to the use of Runge-Kutta explicit schemes by Chima (1986), Moitra et al. (1986), and Jameson et al. (1981). These authors argue that the Runge-Kutta scheme is more efficient than MacCormack's explicit scheme. Further, they point out that a larger time step can successfully be taken with the Runge-Kutta than with the MacCormack approach. However, these Runge-Kutta schemes require the addition of artificial dissipation to remove point-to-point oscillations associated with their use of central differencing for the advective terms. (MacCormack's scheme may also require such a dissipation correction for high Reynolds number flows, though potentially less than the Runge-Kutta scheme due to the one-sided nature of the predictor-corrector scheme.) The Runge-Kutta schemes also require more operational steps per iteration than MacCormack methods.

Explicit methodologies are, of course, not without their inadequacies. An obvious shortcoming of explicit schemes is their

restrictive time step requirement (the "Courant" limitation; see Courant et al. (1929)) which results in a large number of iterations being required to drive a problem to steady-state compared to implicit schemes. However, due to vectorization, the actual amount of computing resources utilized to produce many iterations is much less than for previous scalar machines.

MacCormack (1985) notes that for his explicit predictor-corrector scheme some particular difficulties may arise. The scheme may, for example, never reach machine zero convergence due to the nature of the predictor and corrector steps. Each of these steps is a one-sided finite difference operation (either forward or backward). As a steady-state is approached a point may be reached where the solution oscillates about the steady-state. This problem may be alleviated with an appropriate amount of numerical dissipation.

3.4 STAGGERED VS. REGULAR GRIDS

A third consideration in the solution of the Navier-Stokes equations is the representation of pressure and velocity on a numerical mesh or grid. Two basic types of grid representations exist: regular and staggered. Regular grids are those for which all the flow variables are defined at each of the node points of the grid. In this case, pressure and velocity are defined at the same places on the grid. This type of representation is widely used for compressible flow. Staggered grids are those for which the pressure and velocity values are defined at different grid positions.

The staggered Marker-and-Cell (MAC) grid was first discussed by Harlow and Welch (1965) and was further considered by Hirt and Cook (1972). The classical MAC method springs from these works. This type

of grid, and adaptations thereof for generalized curvilinear coordinate transformations, have found favor with many incompressible flow solvers because they preserve the physical relationship between pressure and velocity better than the regular grid structure for this type of flow. However, several researchers continue to use regular grids for incompressible flow solution (such as Rhie (1986)) in conjunction with various levels of numerical dissipation.

Not withstanding the use of regular grids by some incompressible flow researchers, Bernard and Thompson (1984) have shown that some type of grid staggering is needed for stable solutions to the incompressible Navier-Stokes equations at high Reynolds numbers in the absence of artificial viscosity (the latter being required by many regular grid solutions). These authors further state that, for orthogonal grids, the MAC grid scheme is quite adequate; a recent three-dimensional incompressible flow solution by Cooley (1986), for example, uses the MAC grid. Ferziger (1987) points out the staggered grid has a number of advantages over the regular grid in that the former, by locating flow variables at the centers of control volumes, increases the accuracy of the differencing formula. Further, the staggered grid scheme conserves mass and momentum in a very natural way. In the past, these advantages were offset by the difficulties that staggered grids had when variable grid spacing or boundary-fitted grids are used; however, methods have recently been developed to overcome these difficulties.

3.5 PRIMITIVE VARIABLE SOLUTION VS. OTHER FORMULATIONS

The Navier-Stokes equations, as expressed in their original form, are in primitive variable form. This means that the flow variables will be pressure and velocity. This type of solution is contrasted with

those that solve the governing equations, in a modified form, for vorticity and the streamfunction. Many early numerical solutions used this latter methodology while following the lead of Fromm (1963). The streamfunction-vorticity form of the Navier-Stokes equations can be obtained easily by taking the curl of the governing primitive variable equations. This approach is particularly appealing because , as Orszag and Israeli (1974) state, vorticity is generated locally near boundaries in high Reynolds number flows and is subsequently diffused and advected away. Conversely, pressure is governed by an elliptic equation and is affected instantaneously by all points in the flow domain. The streamfunction-vorticity formulation conserves mass at all points in the flow domain under all circumstances. Roache (1972), in fact, strongly supports this formulation for the solution of the two-dimensional equations of motion. Nonetheless, Orszag and Israeli claim that primitive variable formulations are generally somewhat more accurate than streamfunction-vorticity formulations because the latter formulation requires the numerical approximation to more derivatives than does the primitive variable approach.

The biggest shortcoming of the streamfunction-vorticity formulation is that it requires specification of the streamfunction and vorticity on boundaries. This is very difficult to specify for flow about obstacles such as bridge piers and islands. Assignment of the streamfunction for said piers would be tantamount to assigning the character of the flow in the very region of interest. Three-dimensional generalizations of the streamfunction-vorticity method have been developed using a vector potential (such as Aziz and Hellums (1967) and Hirasaki and Hellums (1970)). These methods, however, are again difficult to formulate at

the flow boundary. Other three-dimensional formulations using a velocity-vorticity method (i.e., Reizes et al. (1984) and Chien and Schetz (1975)) suffer from similar difficulties.

3.6 SOLUTION SCHEME ACCURACY/MASS CONSERVATION

Accuracy is of extreme importance for numerical solution to a system of partial differential equations. With the incompressible Navier-Stokes equations, assurance of mass conservation (i.e., the divergence of velocity is less than one percent for every grid cell) is a fundamental concern. Many numerical schemes attempting solution of the incompressible equations have had success in assuring conservation of momentum. However, many of the same schemes "leak" in that they do not conserve mass locally and/or globally. This lack of mass conservation is equal to allowing compressibility in the incompressible calculation. The effects of compressibility should be kept low so that the basic character of the flow field will be preserved. The acceptance of compressibility introduces a finite sound speed into a flow field that should have an infinite sound speed. If too much compressibility is introduced via mass violations, pressure changes will not be transmitted properly within the flow field.

Bernard and Thompson (1984) point out that the use of central differences for representation of the advective terms in the governing equations generally leads to point-to-point oscillations that contaminate and destabilize flow-field calculations. This is true whether the grid scheme used is staggered or regular. Further, these oscillations, while physically meaningless, do conserve mass and momentum and are, therefore, a solution to the discretized equations in the strictest sense. Several ways are available to mitigate these oscillations

including the use of explicit artificial dissipation and/or upwind differencing. However, Bernard and Thompson (1984) state that the best method to overcome this problem is to use a discretization scheme which ties the solution at each point in the flow field directly to analogous variables at its nearest discrete neighbors. Thus, predictor-corrector schemes, with their alternating use of one-sided differences, help control oscillations. Note also that the use of staggered grids promotes accuracy by representing velocity (or mass flux) and pressure in their proper relationship (finite volume representation) for incompressible flow.

3.7 POISSON VS. CHORIN PRESSURE SOLUTION

Another obstacle to the solution of the incompressible Navier-Stokes equations involves the lack of a time derivative in the continuity equation (which may negate the use of time-marching schemes) and the lack of an obvious equation of state relating pressure to other flow variables such as density. This situation has often been remedied through manipulation of the continuity and momentum equations to produce a Poisson equation relating pressure and the velocity field. The mathematical details of this approach are presented in Chapter 4.

When using the Poisson equation approach, the pressure is obtained from a Poisson equation based on some intermediate velocity field approximation. The velocity field is then modified using the gradients of this new pressure field. Some iteration is necessary at each time step to converge the Poisson solution. The approach was originated by Harlow and Welch (1965), and the basic concept has been subsequently used by many incompressible flow researchers. The basic differences between the treatments presented by these investigators have centered on

two points: (a) simplification of the Poisson equation; and, (b) the iteration scheme used to solve the equation. These points are discussed below.

Patankar and Spalding (1972) developed the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) method for providing the updated pressure field from a Poisson equation. The Poisson equation is simplified by assuming that the influences of the local pressure on other than its nearest neighboring velocity points is negligible. This method is known to be accurate, but it converges slowly. The SIMPLER (SIMPLE Revised) method by Patankar (1981) improves on the SIMPLE method by keeping all of the terms neglected in the latter, with the result that convergence is improved at the cost of additional computational time per iteration. Neely and Claus (1985) present a derivation of these methods which they state has the advantages of both. All of these methods use under-relaxation in their solution for the pressure field and solve a version of the Poisson equation. They differ from each other in the number of terms kept in the Poisson equation and in the exact numerical approach taken for solution.

There are many numerical methods for the solution of the Poisson equation for pressure. Most are iterative due to the prohibitive nature of direct solution for this equation. The methods mentioned above use under-relaxation techniques; Bernard (1988) uses a conjugate-gradient method; Vanka and Misegades (1987) use a vectorizable Gauss-Seidel scheme; and still others have used ADI-like schemes. Of these, the conjugate-gradient method seems to be the most efficient, and it is also vectorizable.

The iterative nature of the majority of Poisson solution methods has led several researchers to question its efficient use for three-dimensional, steady-state simulations. An alternative to such steady-state solution approaches is the introduction of what is called "pseudo-compressibility". This concept, which is roughly equivalent to introducing some compressibility into the incompressible continuity equation through a time derivative for pressure, was first proposed by Chorin (1967). The basic idea of the method is to consider the steady-state incompressible flow solution as the limit as time approaches infinity of the unsteady equations of motion obtained by coupling the unsteady momentum equations with a modified continuity equation. This modification, explained in Chapter 4, involves the addition of a time derivative of pressure divided by a "pseudo-compressibility" coefficient. As the steady-state is reached, this time derivative vanishes and the original incompressible continuity equation is recovered. Thus, at convergence, mass and momentum should be conserved and the governing incompressible equations solved. Further, as noted in Chapter 4, this method is approximately equal to the full Poisson solution.

Soh and Berger (1984) discuss the advantages of using this formulation versus the full Poisson solution: (a) iterative solution of the Poisson equation for pressure is avoided; (b) inclusion of an explicit time derivative of pressure makes it possible to solve all the governing equations by a time-marching technique while leaving the spatial ellipticity of the equations intact; and, (c) the method is easier to program than the iterative Poisson solution technique. It is for reasons such as these that many researchers have used Chorin's method for pressure including Kwak et al. (1984).

Although there are numerous reasons supporting the use of pseudo-compressibility for steady-state incompressible flow solutions, Chang and Kwak (1984) present several consequences of its use. The method introduces a finite sound speed into what should be an infinite sound speed medium. The modified speed of sound is governed by the pseudo-compressibility coefficient. Ideally, this coefficient should be chosen so that the effective Mach number approaches zero. However, the authors note that this action introduces contamination into their approximately-factored numerical solution. If this coefficient is chosen so that the introduced sound waves move too slowly, the flow field development will be incorrect, especially for boundary layers, due to erroneous pressure information. To remedy this, Chang and Kwak recommend a lower bound for the pseudo-compressibility coefficient.

3.8 CONVERGENCE ACCELERATION

Solution of the incompressible Navier-Stokes equations for even simple flow geometries can take many iterations to reach convergence. For example, a two-dimensional potential flow test case for a simple straight channel discretized on a 21-by-21 grid-point mesh takes over 1,000 iterations to converge to a mass violation of less than 10^{-2}. This translates to potentially large expenditures of computer resources for three-dimensional simulations. A number of techniques exist for acceleration of the convergence of these simulations to the steady state including the use of local time stepping, optimum relaxation parameters, grid stretching, and implicit rather than explicit schemes. One of the most promising approaches is the multigrid method. The practical roots of this method stem from the work of Brandt (Stuben and Trottenberg (1982)), who first realized the practicality of such methods. These

authors, as well as Brandt (1977; 1984) describe the details of the multigrid method. An algebraic interpretation of multigrid has been given by McCormick (1982).

The conceptual basis of the multigrid method is quite straightforward: convergence to the steady-state for most problems slows greatly after the first few iterations, thereby prolonging the expenditure of resources. This happens because relaxation techniques eliminate high frequency errors quickly and longer wavelengths (which make up the bulk of the error after the initial iterations of a relaxation scheme) more slowly. Convergence can be accelerated through the use of multiple grids of ever coarser resolution over a given flow domain. Information on each finer grid is transported to the next coarser grid, where the chosen relaxation technique can smooth the errors of shorter wavelength relative to the coarser grid (but longer relative to the finer grid). Corrections on each coarse grid, the number of grids being dictated by the user, are then referred back to the finer grids through some interpolation mechanism. The result of this action is that the longwave error on the fine grid is smoothed on the coarser grids more quickly resulting in accelerated convergence of the solution on the finest grid. More information on this method is available in Chapters 4 and 5 and in the references cited above.

The uses of multigrid, and the types employed in these uses, are very broad indeed. The types attributed to Brandt include the Correction Scheme, Full Approximation Scheme, Full Multigrid Method, and the Algebraic Multigrid Method. Jameson and Yoon (1986), and Martinelli et al. (1986), present a variant of the Brandt multigrid method that may be somewhat easier to program and use than Brandt's full multigrid

scheme. It uses interpolation to transfer all the coarse grid corrections back to the finest grid rather than using interpolation coupled with relaxation. Ni (1982) presents a further use of multiple grids for convergence acceleration for the Euler equations. This method employs a Lax-Wendroff scheme to provide the corrections to the finest grid as computed on the coarser grids. Johnson (1982) has built on Ni's work by simplifying the computations therein and by employing MacCormack's explicit scheme in his computations (Johnson (1983) and Chima and Johnson (1983)). This method, however, looks very much like the correction scheme of Brandt and has not seen much, if any, use for elliptic problems. This may be contrasted with Brandt's full approximation scheme, which has been used to solve elliptic problems (such as the Poisson equation) and hyperbolic and mixed partial differential equations as well.

The global use of multigrid and multiple-grid schemes for convergence acceleration, as presented by many author is, is impressive. McCormick has edited the proceedings of two international multigrid conferences (McCormick and Trottenberg (1983); McCormick (1985)) held at Copper Mountain, CO; a third proceedings of an 1987 conference is being produced presently. Within this third proceedings volume will be a bibliography of over 600 papers dealing with multigrid theory, application, and innovation. These papers cover all types of applications from aerodynamics to oil reservoir simulation to grid generation. Many papers on mathematical theory are also presented. However, less than 25 papers specifically related to incompressible flow simulation are mentioned. And many of these papers are very recent. Authors such as Fuchs and Zhao (1984), Vanka and Misegades (1987) and Rosenfeld and

Israeli (1987) present papers on incompressible Navier-Stokes flow simulation using multigrid techniques. Further, Ferziger (1987) lists multigrid methods specifically as a special tool of choice for convergence acceleration.

Multigrid methods can be vectorized. They can also be used with both explicit and implicit finite difference methods and finite element methods. The stiffness of most incompressible applications demands the use of a technique of this type. Multigrid methods are not without their shortcomings, however, since they seemingly must be "fine tuned" for each application for which they are used.

3.9 SUMMARY OF METHODOLOGY CHOSEN FOR FURTHER EVALUATION

A summary of the points presented above is given below as supporting arguments for the technique to be further evaluated for three-dimensional incompressible flow simulation. The overall strategy includes the following:

1. Finite difference (finite volume) formulation,
2. MacCormack's explicit predictor-corrector solver,
3. Staggered grid,
4. Primitive variables solution,
5. Mass and momentum conservation locally and globally,
6. Pseudo-compressibility used for pressure solution, and
7. Multigrid method for convergence acceleration used.

Given this type of scheme, the governing equations are presented next. The actual numerical discretization of these equations is presented in Chapter 5.

CHAPTER 4

GOVERNING EQUATIONS FOR INCOMPRESSIBLE FLOW

4.1 OVERVIEW

Presented in this chapter are the governing equations for incompressible flow of a Newtonian fluid. The fluid is understood to be homogenous, thereby negating buoyancy effects upon its motion. In addition, the free surface of the fluid is assumed flat (rigid lid concept), an assumption which is usually valid for low Froude number flows such as those approaching hydraulic structures. Given these assumptions, the governing equations expressing conservation of mass and momentum are presented. Note that, in the absence of buoyant body forces, the energy equation is completely uncoupled from the mass and momentum conservation equations.

4.2 CONSERVATION EQUATIONS

The governing equations of motion (Navier-Stokes equations) for incompressible flow, given the above assumptions, in cartesian coordinates and primitive variables are:

$$u_t + (uu)_x + (uv)_y + (uw)_z = -p_x + \nu\nabla^2 u \qquad (4.1)$$

$$v_t + (vu)_x + (vv)_y + (vw)_z = -p_y + \nu\nabla^2 v \qquad (4.2)$$

$$w_t + (wu)_x + (wv)_y + (ww)_z = -p_z + \nu\nabla^2 w \qquad (4.3)$$

In addition, the governing equation of mass conservation is

$$u_x + v_y + w_z = 0 \qquad (4.4)$$

where

x, y, z = cartesian coordinates

u, v, w = cartesian velocity components

ν = kinematic viscosity

∇^2 = laplacian operator

p = dynamic pressure, divided by density

and the subscripts denote partial derivatives of the given index.

Equations 4.1-4.3 are transport equations relating to the conservation of linear momentum. Equation 4.4 represents a constraint that must be satisfied at each instant, that being the constraint of mass conservation both locally and globally. These equations are the Reynolds-averaged equations, and their solution results, naturally, in the computation of mean velocities and pressure. It should be noted that the pressure in these equations is actually the dynamic component of pressure and does not include the hydrostatic component. This presents no problem given that it is the pressure gradient, and not the actual pressure, which is of prime import due to its role as an agent of mass conservation in incompressible flow.

Bernard (1986) has shown that it is often advantageous to express the advective terms in Equations 4.1-4.3 in an asymptotically conservative form, thereby effectively adding a term of the form

$$u\,(u_x + v_y + w_z)$$

(or its analogy) to the right-hand sides of each equation of motion. Bernard notes that this formulation eliminates false source terms that might enter into the momentum equations (equations of motion) as a result of mass not being conserved through the chosen numerical solution scheme.

In order to more accurately and efficiently provide computations in regions with arbitrary geometries, such as in the vicinity of irregular boundaries, a coordinate transformation is introduced which maps the solution of the governing equations from a nonuniform physical domain to a rectangular computational space. Thus, it is advantageous to write the governing equations in terms of the generalized curvilinear coordinates

$$\xi = \xi(x,y)$$

$$\eta = \eta(x,y)$$

Given this transformation of coordinates, the two-dimensional analog of Equations 4.1-4.4 above become

$$\frac{u_t}{J} + (uU)_\xi + (uV)_\eta = \frac{-p_x}{J} + \frac{\nu\nabla^2 u}{J} + u(U_\xi + V_\eta) \qquad (4.5)$$

$$\frac{v_t}{J} + (vU)_\xi + (vV)_\eta = \frac{-p_y}{J} + \frac{\nu\nabla^2 v}{J} + v(U_\xi + V_\eta) \qquad (4.6)$$

$$(U_\xi + V_\eta) = 0 \qquad (4.7)$$

where the quantities U and V are volumetric flux components and J is the Jacobian of the transformation as given by

$$U = y_\eta u - x_\eta v \tag{4.8}$$

$$V = x_\xi v - y_\xi u \tag{4.9}$$

$$J = (x_\xi y_\eta - y_\xi x_\eta)^{-1} \tag{4.10}$$

where

x_ξ, x_η, y_ξ, y_η are the metrics of the transformation whose numerical formulations are defined in Chapter 5.

Note that J is actually the inverse of the volume of a given grid cell (control volume) in the computational plane. Only the equations of motion in two dimensions are presented since the bulk of this report will consider their numerical solution and the implications of their solution for three-dimensional solutions. All additional presentations of the governing equations will be in two dimensions.

The cartesian velocity components, u and v, may also be expressed as functions of the flux components shown above.

$$u = J(x_\xi U + x_\eta V) \tag{4.11}$$

$$v = J(y_\xi U + y_\eta V) \tag{4.12}$$

Further, the first cartesian derivatives for any function f may be expressed, via the chain rule, in the following transformed form

$$f_x = J(y_\eta f_\xi - y_\xi f_\eta) \tag{4.13}$$

$$f_y = J(x_\xi f_\eta - x_\eta f_\xi) \tag{4.14}$$

which results from the fact that, in two dimensions,

$$\xi_x = Jy_\eta \tag{4.15}$$

$$\xi_y = -Jx_\eta \tag{4.16}$$

$$\eta_x = -Jy_\xi \tag{4.17}$$

$$\eta_y = Jx_\xi \tag{4.18}$$

The relationships in Equations 4.13 and 4.14 can then be used to develop a transformed expression for the divergence of a gradient (such being the Laplacian of a given function and the mathematical representation of the viscous terms)

$$\text{div(grad f)} = \nabla^2 f = J\frac{\partial}{\partial \xi}(y_\eta f_x - x_\eta f_y) + J\frac{\partial}{\partial \eta}(x_\xi f_y - y_\xi f_x) \tag{4.19}$$

where the cartesian derivatives of the function f are retained strictly for convenience. Equation 4.19 is then used with the velocity terms, u and v, to compute the viscous terms in the governing equations.

4.3 PRESSURE SOLUTION EQUATIONS

The incompressible equations of momentum and mass conservation, as presented above, are elliptic-parabolic partial differential equations. This contrasts with the compressible equations of motion which are hyperbolic. A characteristic of the relations important in the solution of the incompressible equations is illustrated in Equation 4.7 by the absence of both a time derivative for either density or pressure and an advective term for the gradient of the same. In addition, there is no equation of state relating pressure and density for an incompressible flow as there is for compressible flow. Thus, methodologies must be developed which allow for the mathematical coupling of the pressure and velocity fields. Two primary approaches exist, as discussed in Chapter 3, to express pressure as a function of the velocity field: (a) the solution of a Poisson equation for pressure; and (b) the use of pseudo-compressibility. As will be shown in this section, (b) is actually a specialized version of (a). The foundation of each of these methods will be presented below.

4.3.1 Poisson Pressure Formulation

To illustrate the formulation of the Poisson equation relating pressure and the velocity field, recall the equations of motion in primitive vector form

$$\underset{\sim}{u}_t + \underset{\sim}{u} \cdot \nabla \underset{\sim}{u} + \nabla p = \nu \nabla^2 \underset{\sim}{u} \tag{4.20}$$

where $\underset{\sim}{u}$ is the vector of velocities and ∇ is the gradient operator.

As a first step in this development, use a simple two-point time-differencing scheme for the time derivative of velocity such that

$$\underset{\sim}{u}^{n+1} + \Delta t \nabla p = \underset{\sim}{u}^{n} - \Delta t \left[\underset{\sim}{u}^{n} \cdot \nabla \underset{\sim}{u}^{n} - \nu \nabla^2 \underset{\sim}{u}^{n}\right] \tag{4.21}$$

Equation 4.21 can be simplified by using the following definitions:

$$\phi = p\Delta t \tag{4.22}$$

$$\underset{\sim}{f}^{n} = \underset{\sim}{u}^{n} - \Delta t \left[\underset{\sim}{u}^{n} \cdot \nabla \underset{\sim}{u}^{n} - \nu \nabla^2 \underset{\sim}{u}^{n}\right] \tag{4.23}$$

Thus,

$$\underset{\sim}{u}^{n+1} + \nabla \phi = \underset{\sim}{f}^{n} \tag{4.24}$$

Now, taking the divergence of Equation 4.24, and recalling that the desired pressure is that which results in a divergence-free velocity field (mass is conserved) at the time level n+1, the following Poisson equation results

$$\nabla^2 \phi = \nabla \cdot \underset{\sim}{f}^{n} \tag{4.25}$$

Recalling that the vector f is one whose components are estimates of the new nonconservative velocity components at the time step n + 1, the divergence of f can be defined as

$$\nabla \cdot \underset{\sim}{f}^{n} = \nabla \cdot \underset{\sim}{u}' \tag{4.26}$$

where $\underset{\sim}{u}'$ is the provisional velocity defined by the right-hand side of Equation 4.23. Thus, the pressure that is required at the new time step

to insure mass conservation is related to a provisional velocity field whose divergence may be nonzero (and thus, does not conserve mass). Equation 4.25 is a Poisson equation which is usually solved by iterative means. As was discussed in Chapter 3, many researchers have questioned the efficiency of solving this equation by iteration at each time step within the framework of a global solution scheme which is, itself, iterative. These researchers have then often opted for use of the method of pseudo-compressibility as initiated by Chorin (1967) for steady-state computations.

4.3.2 Pseudo-Compressibility Formulation

Suppose, in the solution of the governing conservation equations, that an explicit iterative scheme were used. Given the definitions above for ϕ and the vector f, Equation 4.24 could then be replaced by

$$\underset{\sim}{u}^{m} + \nabla\phi^{m} = \underset{\sim}{f}^{m-1} \qquad (4.27)$$

where m denotes the iteration count. It is necessary, as Equation 4.27 converges, that the divergence of the velocity field approaches zero. However, there is no direct mechanism at each iteration to ensure this action.

In order to ensure a divergence-free field at convergence, consider the solution of Equation 4.27 through some iteration scheme. As the solution approaches convergence, the left-hand side of Equation 4.27 approaches a steady-state such that

$$\underset{\sim}{f}^{m} = \underset{\sim}{f}^{m-1} \qquad (4.28a)$$

$$\underset{\sim}{u}^{m} + \nabla\phi^{m} = \underset{\sim}{u}^{m-1} + \nabla\phi^{m-1} \tag{4.28b}$$

Taking the divergence of Equation 4.28b, and requiring that the velocity field at iterate m be divergence-free, one obtains

$$\nabla^2\phi^{m} = \nabla \cdot \underset{\sim}{u}^{m-1} + \nabla^2\phi^{m-1} \tag{4.29}$$

Thus, it appears that the Poisson Equation 4.29 must be solved exactly for each sweep of the field given the known old pressure and velocity fields. However, Equation 4.29 itself can be used to define an alternative iteration solution procedure for ϕ by employing successive over-relaxation (SOR),

$$\nabla^2\phi^{*} = \nabla \cdot \underset{\sim}{u}^{m-1} + \nabla^2\phi^{m-1} \tag{4.30}$$

$$\phi^{m} = \omega\phi^{*} + (1-\omega)\phi^{m-1} \tag{4.31}$$

where ω is a relaxation parameter.

The iteration scheme above is still no easier to employ than the Poisson pressure solution discussed at the beginning of this section. However, simplification of this alternative scheme is straightforward. Recall that the Laplacian in Equation 4.30 can be expressed by the five-point scheme

$$\nabla^2\phi = \frac{1}{\Delta x^2}\left(\phi_{i+1j} - 2\phi_{ij} + \phi_{i-1j}\right) + \frac{1}{\Delta y^2}\left(\phi_{ij+1} - 2\phi_{ij} + \phi_{ij-1}\right) \tag{4.32}$$

Equation 4.32 is first substituted into Equation 4.30. The non-main diagonal elements of ϕ^* and ϕ^{m-1} are then equated in this new expression. Assuming the spatial step sizes are equal for convenience, this operation yields

$$\phi^m = \phi^{m-1} - \frac{\Delta x^2 \omega}{4} (\nabla \cdot \underset{\sim}{u}^{m-1}) \tag{4.33}$$

Recalling the definition of phi, this yields

$$p_{ij}^m = p_{ij}^{m-1} - \frac{\Delta x^2 \omega}{4\Delta t} (\nabla \cdot \underset{\sim}{u}^{m-1}) \tag{4.34}$$

Rearranging (4.34) and dividing both sides by the time step gives

$$\frac{p_{ij}^m - p_{ij}^{m-1}}{\Delta t} + \frac{\Delta x^2 \omega}{4\Delta t^2} (\nabla \cdot \underset{\sim}{u}^{m-1}) = 0 \tag{4.35}$$

which is the explicit, first-order (in-time) approximation for the equation

$$\frac{\partial p}{\partial t} + \beta^2 (\nabla \cdot \underset{\sim}{u}) = 0 \tag{4.36}$$

where

$$\beta^2 = \frac{\Delta x^2 \omega}{4\Delta t^2} \tag{4.37}$$

Equation 4.34 is solvable directly for the new pressure for each control volume as a function of the old pressure and velocity fields directly. Further evaluation of the method shows that the coefficient β acts as a "pseudo" sound speed which is introduced by numerical contrivance alone. The physical sound speed is defined as

$$\bar{c} = \left(\frac{dp}{d\rho}\right)^{1/2} \tag{4.38}$$

at constant entropy. Thus, if pressure (here the actual pressure rather than the pressure divided by the density as utilized elsewhere) were a function of density alone, and given that β is acting as a surrogate sound speed, one could write

$$\frac{\partial p}{\partial t} = \frac{dp}{d\rho}\frac{\partial \rho}{\partial t} = \beta^2 \frac{\partial \rho}{\partial t} \tag{4.39}$$

which, when substituted into Equation 4.36 yields

$$\beta^2 \frac{\partial \rho}{\partial t} + \rho\beta^2 (\nabla \cdot \underset{\sim}{u}) = 0 \tag{4.40}$$

Therefore, this method suggested by Chorin is nothing more than an acoustic approximation to the compressible continuity equation. Given the absence of an advective term for the density gradient in Equation 4.40, which would appear in the compressible equation, Equation 4.36 is valid only when the density gradients in a given flow are very small; i.e., when true compressibility is small, such as for low Froude-number incompressible flows.

From Equation 4.40 the origin of the name of this method, "pseudo-compressibility" can be easily seen. At each time step or iteration, the solution of Equation 4.40 (or, more straightforwardly, Equation 4.34 for pressure) is tantamount to accepting some amount of "compressibility" in the incompressible flow solution. However, at the steady state, the solution of Equation 4.34 should return the required condition that the velocity field be divergence-free (Equation 4.7). Use of the method may return some degree of hyperbolicity to the governing equations prior to the steady-state. However, as the steady-state is approached, the governing equations must return to their elliptic nature for proper solution. The true advantage of the use of this approach is that marching schemes can be employed within which no additional iteration is required for pressure solution as would be the case with a Poisson pressure solution. Use of this method, however, places some stability limits on the calculations (as discussed in Chapter 6). Further, it does not, unto itself, guarantee the computation of a pressure field at each iteration which will ensure mass conservation in contrast to the Poisson solution discussed above.

4.4 CONVERGENCE ACCELERATION METHODOLOGY

Chapter 7 describes a solution of the two-dimensional equations of motion with an explicit marching scheme employing pseudo-compressibility which requires many thousand iterations to achieve a steady state. A three-dimensional calculation would require many more. To make these calculations more attractive fiscally, an efficient method is sought to accelerate the convergence of these calculations to the steady state. Chapter 3 concluded that the most attractive convergence acceleration methodology for solution of partial differential equations is the

multigrid approach inspired by Brandt (1984). The actual multigrid scheme utilized herein, which is founded on the work of Brandt (1984) and Jameson and Yoon (1986), is described below for the solution of a general system of partial differential equations. The mechanics of this solution are presented in Chapter 5.

4.4.1 Approach Heuristics

The explanation of the multigrid approach is begun by expressing the Navier-Stokes Equations (4.1-4.4) as a partial differential system designated by

$$L\underset{\sim}{u} = \underset{\sim}{f} \tag{4.41}$$

where L is a differential operator and, in this case, nonlinear. A numerical solution to this equation can be constructed through the discretization of the solution space and the formulation of L as the discrete operator L'. The numerical approximation to Equation 4.41 would then be

$$L'\underset{\sim}{u}' = \underset{\sim}{f}' \tag{4.42}$$

where the primes all denote a numerical approximation. In general, some error will be incurred in the numerical approximation of the $\underset{\sim}{u}$ such that

$$\underset{\sim}{u} = \underset{\sim}{u}' + \underset{\sim}{\varepsilon} \tag{4.43}$$

where $\underset{\sim}{\varepsilon}$ denotes the error associated with the unconverged numerical approximation to $\underset{\sim}{u}$. It is this error $\underset{\sim}{\varepsilon}$ which must be reduced to acceptable tolerances before the steady-state solution can be achieved.

The reduction of the error $\underset{\sim}{\varepsilon}$ for the Navier-Stokes equations is not a simple task. Firstly, $\underset{\sim}{\varepsilon}$ has within it various wavelengths of error, some short and some quite long. It is well known that relaxation techniques are quite good at smoothing error bands which are of high frequency/short wavelength. These wavelengths are generally on the order of the grid spacing. However, the same techniques are generally not as good at resolving the longer wavelength errors. Thus, it is these long wavelength errors that result in protracted iteration toward the steady-state solution.

Multigrid techniques seek to enhance convergence by smoothing errors of longer wavelengths as well as the high frequency error. This is done through the use of a number of grids of increasing coarseness (i.e., decreasing number of grid points) as compared to the primary solution space discretization (hereafter referred to as the finest grid). To each of these coarse grids information from the previously finer grid is transferred. The relaxation scheme of interest is then applied on this coarser grid, thereby smoothing the short wavelengths of error on said grid as discussed in Chapter 3. While the multigrid concept may appear simple, the actual mechanisms associated with implementing it correctly are somewhat more difficult. These steps are overviewed below.

4.4.2 Steps for Multigrid Implementation

There are three main steps in a multigrid calculation: (a) restriction; (b) relaxation; and (c) prolongation. Each of these steps

may be applied numerous times during a given global solution. Given below is an overview of the steps taken to implement multigrid.

4.4.2.1 Step 1. Following the initialization of the solution domain, including grid generation and data input, one or more relaxation sweeps are performed on the finest grid (grid k) using the MacCormack explicit predictor-corrector scheme (discussed in Chapter 5). From this, an updated approximate solution is generated. If this solution meets some specified criteria for convergence, calculation ceases and the solution is accepted as the approximation to the steady state.

4.4.2.1 Step 2. Assuming convergence has not been reached, the updated solution is transferred to the next coarser grid (grid k+1). This grid can have any spatial increment; however, a computationally convenient coarse grid spacing is twice that of the previously finer grid, and this convention is used herein. The updated flux and pressure gradient information from the finer grid is transferred to the coarser grid via a process generally referred to as restriction. Although a number of restriction operators exist, integration based on the Gauss Divergence Theorem (Kreyszig, 1979) is employed herein. The mechanics of this operation are detailed in Chapter 5.

4.4.2.3 Step 3. In a manner analogous to Step 2, the finer grid residuals are restricted to the coarser grid. These residuals are defined as

$$\underset{\sim}{r}^{'k} = \underset{\sim}{f}^{'k} - L^{'k}\underset{\sim}{u}^{'k} \qquad (4.44)$$

where

$\underset{\sim}{r}^{'k}$ = present residuals on finer grid k

$\underset{\sim}{f}'^{k}$ = present right-hand-sides of differential system being solved on grid k

$\underset{\sim}{u}'^{k}$ = present solution approximation on finer grid k

4.4.2.4 Step 4. Recalling that the purpose of the multigrid approach is to accelerate finest grid solution convergence via error smoothing, one would now like to solve directly for this error on the coarser grid. Due to the nonlinear nature of the Navier-Stokes equations, however, the L' does not operate directly upon $\underset{\sim}{\varepsilon}$ but upon its alternate expression $\underset{\sim}{u} - \underset{\sim}{u}'$. Then, employing Brandt's (1984) Full Approximation Scheme (FAS), the following system is relaxed on grid k+1

$$L'^{k+1}\,\underset{\sim}{u}'^{k+1} = L'^{k+1}\,(I'^{k+1}_{k}\,\underset{\sim}{u}'^{k}) + I'^{k+1}_{k}\,\underset{\sim}{r}'^{k} \qquad (4.45)$$
$$= \underset{\sim}{f}^{k+1}$$

where

$\underset{\sim}{u}'^{k+1}$ = updated solution approximation on coarser grid k+1

I'^{k+1}_{k} = restriction operator defining the finer-to-coarser grid transfers of information

L'^{k+1} = finite difference approximation to L'^{k} on coarser grid k+1

and all other variables are as defined above.

The finite difference operator on the coarser grid is exactly the same as that on the finest grid. Thus, MacCormack's predictor-corrector scheme is employed to solve Equation 4.45 for a new estimate of the vector $\underset{\sim}{u}$ on the coarser grid. The restriction operator I'^{k+1}_{k} will be defined in Chapter 5.

Equation 4.45 can be expressed in words as follows: the new approximate solution for the governing system of partial differential equations can be solved for on the coarser grid k+1 using the same relaxation scheme as was used on the finest grid (L'^{k}). This scheme operates first on the transferred (restricted) current finer-grid solution ($\underset{\sim}{u}'^{k}$) whose value is based on its restriction to grid k+1 through the transfer operator I'^{k+1}. This coarse grid solution is further refined based on the residuals computed and transferred from the finer grid, $\underset{\sim}{r}'^{k}$. The results of these two operations are summed and serve as the new right-hand side of the governing system of equations being solved. The basic relaxation scheme then operates on this new system.

4.4.2.5 Step 5. Repeat Steps 2-4, transferring the new coarse solution on grid k+1 to a still coarser grid (note that the resolution on this grid is one-fourth as fine as on grid k) based on the operations described in Step 2. Repeat these operations for as many coarse grids (N) as prescribed by the user. In this fashion, the residuals from the previous grid drive the solution on the next coarser grid.

4.4.2.6 Step 6. Having completed restriction and relaxation operations on each coarse grid, transfer the coarse corrections to the approximate solutions for each just finer grid (and, subsequently, the finest grid) through a process called prolongation. The corrections are prolonged (transferred) from grid h+1 (a coarser grid) to grid h (next finer grid) based on the equation (Brandt, (1984))

$$\underset{\sim}{u}^{,h}_{new} = \underset{\sim}{u}^{,h} + I^{,h}_{h+1}\left(\underset{\sim}{u}^{,h+1} - I^{,h+1}_{h}\ \underset{\sim}{u}^{,h}\right) \qquad (4.46)$$

where the $\underset{\sim}{u}'$ are the current values of the approximate solution on grids

h and h+1; $\underset{\sim}{u}'_{new}$ the updated approximate solution on finer grid h, and the I' transfer operators. Note that I'^{h+1}_{h} is actually the same as the restriction operator described above for Equation 4.45. I'^{h}_{h+1}, conversely, is the prolongation operator which defines the transfer of coarse grid corrections to a finer grid. This operator is also defined in Chapter 5.

Note in Equation 4.46 that only the correction to the coarse grid approximate solution is prolonged from grid h+1 to grid h. This is shown by the subtraction from the current solution on grid h+1, which was solved for in Equation 4.45, of the initial solution on said grid as transferred from grid h in the restriction phase of Step 2.

4.4.2.7 Step 7. Repeat Step 7 for all grids, culminating in the transfer of global coarse-grid corrections from all the coarser grids to the finest grid (grid k). The value of u'^{k}_{new} is then the value of $\underset{\sim}{u}$ at the new time step.

4.4.2.8 Step 8. Repeat the entire process, starting with Step 1, for a pre-specified number of multigrid cycles or until convergence is reached.

This procedure is similar to the V-cycle Full Multigrid approach discussed by Brandt (1984). Within the prolongation process (Steps 6 and 7), Brandt first transfers coarser-grid connections to the next finer-grid, then relaxes this modified finer-grid solution. This, in turn, produces updated corrections for prolongation to the next level of finer grid. The procedure used herein performs no such additional relaxation within the prolongation process, choosing rather to prolong via interpolation (discussed in Chapter 5) alone. This is similar to the procedure of Jameson and Yoon (1986). This approach was chosen based on its apparent simplicity compared to Brandt's scheme.

CHAPTER 5

NUMERICAL SOLUTION METHODOLOGY

5.1 INTRODUCTION

In order to simulate incompressible flows by means of a numerical model, one must discretize the governing equations and program the rules presented in Chapter 4. This chapter presents the methodology employed for this discretization and the numerical solution to the incompressible Navier-Stokes equations. The chapter begins with a detailed explanation of the use of the predictor-corrector scheme of MacCormack (1969) to solve the governing equations of mass and momentum conservation in generalized two-dimensional curvilinear coordinate space. The incorporation of a multigrid scheme for convergence acceleration into the solution scheme is then discussed. As the heuristics of this scheme were presented in Chapter 4, the interworkings of the transfer mechanisms, restriction and prolongation, and the computation of the residuals will be detailed herein. Discussion of boundary conditions, initial conditions, stability calculations, and convergence criteria is presented in the next chapter.

5.2 PROBLEM DISCRETIZATION

Discretization involves the division of the flow field into a finite number of individual cells whose boundaries may be permeable (such as for inlets or outlets) or impermeable (solid walls). The congregation of cells in this domain constitutes a finite volume grid

that represents the physical domain. This domain may have any arbitrary shape in the physical (cartesian) plane, such as in Figure 5.1. However, it is transformed as discussed in Chapter 4 into a

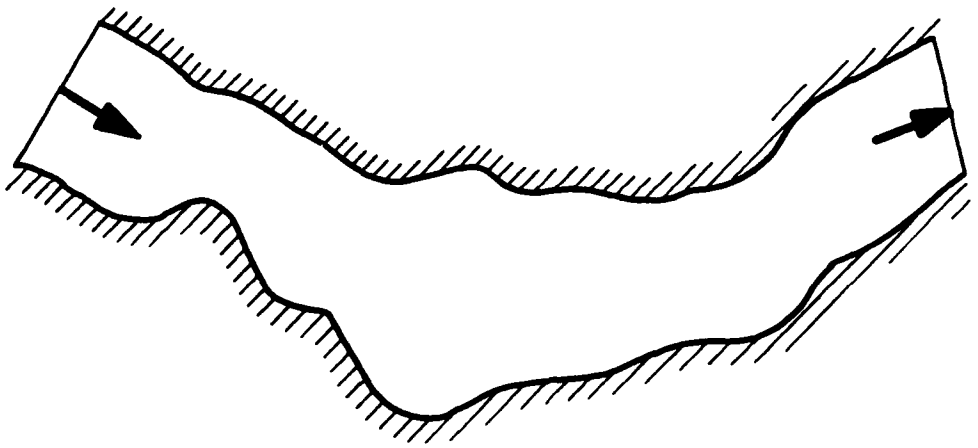


Figure 5.1. Arbitrary physical domain

rectangle in computational (ξ,η) space as shown in Figure 5.2. The transformation from physical to computational coordinates carries all information pertinent to grid spacing in the physical plane; thus, the computational grid spacing has no effect on the physical results. The spacing is therefore chosen for convenience to be the following:

$$\Delta\xi = \Delta\eta = 1 \tag{5.1}$$

This results in a grid which, if orthogonal, is perfectly square in the computational plane. It is in this plane that all calculations are carried out.

The quantities reflecting velocity (u and v) and pressure (p) are defined at the center of each grid cell as shown in Figure 5.2. The flux U at point (i,j) is defined at the midpoint of the left (west) face, and the flux V at the midpoint of the lower (south) face. Each cell is identified through an integer indexing system (i,j) such that

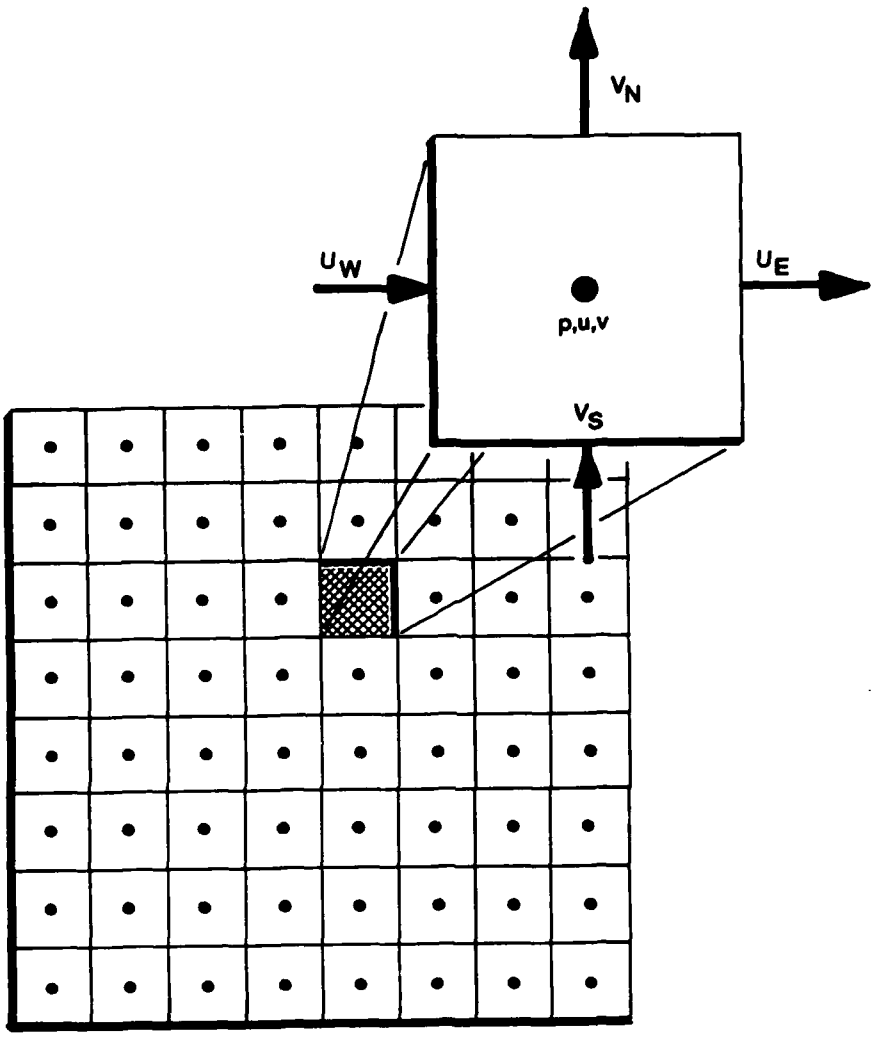


Figure 5.2. Computational domain with variable definition

the cell center coincides with a position in the computational plane given by

$$\xi = i + 1/2 \tag{5.2}$$

$$\eta = j + 1/2 \tag{5.3}$$

The values of x and y are specified at the grid cell corners such that x(i,j) and y (i,j) are located at the lower left (southwest) corner of

each (i,j) cell. Defining (e,w,n,s) to denote evaluation of certain quantities or fluxes on the east, west, north, and south faces, respectively, of the (i,j) cell, and allowing c to represent the center point in the given cell, the metrics defining the changes in the x-axis with respect to the computational coordinates are

$$x_{\xi}^{n} = x(i+1,j+1) - x(i,j+1) \tag{5.4}$$

$$x_{\eta}^{e} = x(i+1,j+1) - x(i+1,j) \tag{5.5}$$

$$x_{\xi}^{e} = 0.25\,[x(i+2,j+1) - x(i,j+1) + x(i+2,j) - x(i,j)] \tag{5.6}$$

$$x_{\eta}^{n} = 0.25\,[x(i+1,j+2) - x(i+1,j) + x(i,j+2) - x(i,j)] \tag{5.7}$$

$$x_{\xi}^{c} = 0.50\,[x(i+1,j+1) - x(i,j+1) + x(i+1,j) - x(i,j)] \tag{5.8}$$

$$x_{\eta}^{c} = 0.50\,[x(i+1,j+1) - x(i+1,j) + x(i,j+1) - x(i,j)] \tag{5.9}$$

Similar expressions apply for the y-coordinate metrics. Since the north and east faces of cell (i,j) correspond to the south and west faces of cells (i,j+1) and (i+1,j), respectively, these expressions also provide for the metrics with respect to these latter faces as well. Only certain of these expressions are required, however, for cells having common faces with boundaries. Vertical boundaries will require computation of those metrics which are with respect to the η coordinate since only those are needed to construct the U fluxes; conversely, only ξ-related metrics are required on the horizontal boundaries. Thus, the

required metrics for a vertical boundary are defined as

$$x_\eta = x(i,j+1) - x(i,j) \tag{5.10}$$

while for a horizontal boundary

$$x_\xi = x(i+1,j) - x(i,j) \tag{5.11}$$

where analogous expressions exist for the y-coordinate metrics.

Given that the velocities (u,v) and fluxes (U,V) are defined at different locations for a given cell, it is useful to introduce the shift indices (r,s) which can be used to relate velocity and flux components on the staggered grid cell shown in Figure 5.2. The shift indices are pairs of integers with values of either zero or one in each of the four possible combinations of the two. For example,

$$(r,s) = (1,0)$$

These pairs are changed systematically (such as every iteration) to allow for a more symmetric computation in the predictor-corrector scheme and to maintain computational stability.

For convenience, all quantities will be assumed to have the indices (i,j) unless specifically stated otherwise. The fluxes through the east, west, north, and south faces of each cell are U (i+1,j), U (i,j), V (i,j+1), and V (i,j), respectively. Combining the shift indices with Equations 4.11 and 4.12, the following relationships now are defined

between the cell-centered velocity components and the face-centered flux components:

$$u(i,j) = J_c \left[x_\xi^c (i,j)\, U(i+r,j) + x_\eta^c (i,j)\, V(i,j+s) \right] \quad (5.13)$$

$$v(i,j) = J_c \left[y_\xi^c (i,j)\, U(i+r,j) + y_\eta^c (i,j)\, V(i,j+s) \right] \quad (5.14)$$

The r and s indices thus dictate whether these velocities will be related to the U taken from the east ($r = 1$) face or the west ($r = 0$) face. Similarly, V is taken either from the north ($s = 1$) or south ($s = 0$) face.

For the numerical method presented herein, the cell-centered velocities will always be computed from existing fluxes using the space-shifting relations given by these equations. These velocity components will be used then to find the cell-centered incremental velocity changes in time. The velocity increments are then used to calculate flux incremental changes on the cell faces by reversing the shift operation given above such that

$$\Delta U(i+r,j) = [y_\eta^c\, \Delta u(i,j) - x_\eta^c\, \Delta v(i,j)] \quad (5.15)$$

$$\Delta V(i,j+s) = [x_\xi^c\, \Delta v(i,j) - y_\xi^c\, \Delta u(i,j)] \quad (5.16)$$

Prior to presenting details of the predictor-corrector scheme, it will prove useful to introduce the difference operators

$$D_\xi (Uf)_{ij} = U(i+1,j)\, f(i+r,j) - U(i,j)\, f(i+r-1,j) \quad (5.17)$$

$$D_\eta (Vf)_{ij} = V(i,j+1)\ f(i,j+s) - V(i,j)\ f(i,j+s-1) \qquad (5.18)$$

where f can represent either of the velocities u and v . Employing the shift indices only in the advective terms, the transport equations for momentum (Equations 4.5 and 4.6) take the discrete form

$$\frac{\Delta f}{J_c \Delta t} + D_\xi(Uf)_{i,j} + D_\eta(Vf)_{i,j} = f(\ D_\xi(U)_{i,j} + D_\eta(V)_{i,j})$$

$$+ \nu \frac{\nabla \cdot (\nabla f)}{J_c} \qquad (5.19)$$

Using the difference operators given in Equations 5.16 and 5.17, and letting the f function be identical to unity, the continuity Equation 4.7 becomes (with pseudo-compressibility incorporated)

$$\frac{\Delta \rho}{J_c \Delta t} + \beta^2 (D_\xi(U)_{ij} + D_\eta\ (V)_{ij}) = 0 \qquad (5.20)$$

5.3 PRIMARY SOLUTION STEPS

Given a set of shift indices (r, s), the predictor-corrector scheme advances the fluxes U and V by one time increment (Δt) in the following manner:

5.3.1 Step 1.

Use Equations 5.13 and 5.14 to compute u and v from existing values of U and V .

5.3.2 Step 2.

Use Equation 5.19 to compute Δu and Δv .

5.3.3 Step 3.

Calculate the flux increments ΔU and ΔV from Equations 5.15 and 5.16, respectively.

5.3.4 Step 4.

Compute new fluxes U and V by adding the respective flux increments from Step 3 to the existing fluxes.

5.3.5 Step 5.

Adjust these newer U and V values by subtracting the existing pressure gradient which, as yet, has not been updated.

5.3.6 Step 6.

Use Equation 5.20 to compute the incremental change with respect to time in the pressure as a function of the adjusted fluxes from Step 5.

5.3.7 Step 7.

Adjust U and V a second time to promote conservation of mass by subtracting the gradient of the pressure change computed in Step 6.

In order to accomplish Step 5, it is convenient to introduce the scalar potential

$$\phi = p\ \Delta t \tag{5.21}$$

Letting U' and V' be the non-mass conserving fluxes computed in Step 4, their improved (with respect to mass conservation) components are found from the relations

$$U = U' - (y_{\eta}^{W}\ \phi_{x}^{W} - x_{\eta}^{W}\ \phi_{y}^{W}) \tag{5.22}$$

$$V = V' - (x_{\xi}^{S}\ \phi_{y}^{S} - y_{\xi}^{S}\ \phi_{x}^{S}) \tag{5.23}$$

where the superscripts s and w denote derivatives taken on the south and west faces, respectively. When the derivatives ϕ_x and ϕ_y are evaluated by the chain rule (as shown in Equations 4.13 and 4.14) the results are

$$\phi_x = J\,(y_\eta\,\phi_\xi - y_\xi\,\phi_\eta) \tag{5.24}$$

$$\phi_y = J\,(x_\xi\,\phi_\eta - x_\eta\phi_\xi) \tag{5.25}$$

In order to accomplish Step 7, the scalar potential is redefined as

$$\phi' = \Delta p \Delta t \tag{5.26}$$

where Δp is computed via Equation 5.20. The fluxes are then readjusted via Equations 5.22 and 5.23. In this manner, this solution scheme uses pseudo-compressibility to approximate as closely as possible the changes in the fluxes which would have been dictated by the gradient of a pressure field obtained through a Poisson equation solution. The Poisson solution would have returned the exact pressure field that would have been required to conserve mass for the given provisional fluxes. This pressure field can be thought of as the pressure field that needs to exist such that its gradient, when applied to the existing flux field, will result in perfect mass conservation. The pseudo-compressibility scheme outlined herein, while not as robust, first applies the existing pressure gradient, then adjusts the pressure field given the provisional fluxes and applies the gradient of the changes in said pressure field. Initial simulations have shown this scheme

converges more quickly than a single adjustment of the fluxes based on the existing pressure gradients at the old time step.

The seven-step time marching procedure presented above is suitable for implementation in a two-phase predictor-corrector scheme such as that of MacCormack (1969). In the predictor phase, the seven steps are followed as prescribed, using the existing values of U and V from the previous time step n to calculate the provisional time-advanced flux values with the provisional increments ΔU^n and ΔV^n given below.

$$U^* = U^n + \Delta U^n - (y_\eta \phi_x^n - x_\eta \phi_x^n) - (y_\eta \phi_x'^n - x_\eta \phi_y'^n) \qquad (5.27)$$

$$V^* = V^n + \Delta V^n - (x_\xi \phi_y^n - y_\xi \phi_x^n) - (x_\xi \phi_y'^n - y_\xi \phi_x'^n) \qquad (5.28)$$

The superscript n indicates the time step the solution is being advanced from and the asterisk $*$ indicates the provisionally advanced time level. The constructs given in Equations 5.27 and 5.28 represent the completion of the predictor phase of the scheme.

The corrector phase begins by first changing the shift indices from (r,s) to (r^*,s^*) such that

$$r^* = 1 - r \qquad (5.29)$$

$$s^* = 1 - s \qquad (5.30)$$

The seven steps given above are then repeated exactly as in the predictor phase, using U^* and V^* (rather than U^n and V^n) to compute

the non-conservative flux increments ΔU^{*} and ΔV^{*}. This results in new provisional fluxes U^{**} and V^{**} which are then used in Step 6 to compute the incremental changes in the original pressures resulting from the corrector fluxes. These pressure changes are then used in Step 7 to compute new corrector provisional fluxes that should more closely conserve mass. These new provisional fluxes, U^{***} and V^{***} (from Step 7 of the corrector phase) are then used along with original flux information at time step n to calculate the new fluxes at time step $n+1$ as follows:

$$U^{n+1} = 0.50\ (U^{n} + U^{***}) \tag{5.31}$$

$$V^{n+1} = 0.50\ (V^{n} + V^{***}) \tag{5.32}$$

The new pressure at time step $n+1$ is defined as

$$p^{n+1} = 0.50\ (2p^{n} + \Delta p^{*} + \Delta p^{**}) \tag{5.33}$$

where the Δp terms are the pressure changes from the old pressures as computed from Step 6 of the predictor (*) and corrector (**) phases, respectively.

The predictor-corrector scheme marches the fluxes through time to the steady-state. The shift indices (r,s) must be cycled so that each of the four possible combinations they may have is employed with equal frequency. This is required to remove the asymmetric error and possible instabilities introduced by use of only one combination of these indices. Only the fluxes and pressure values are required for each

calculation; the velocities are obtained from the known grid geometry and the fluxes based on relationships presented above.

It remains to present the discrete numerical representation of the $\nabla \cdot (\nabla f)$ operation that appears in the viscous terms of the momentum equations. This operation is, in the absence of a free surface, simply the Laplacian of the function f (where f represents either the u or v velocity for respective momentum equations). The cartesian form of this operator is very simple; however, the generalized curvilinear form is much more involved. Therefore, the generalized curvilinear form of the Laplacian is presented in Appendix B.

5.4 INCORPORATION OF MULTIGRID METHODOLOGY

There are several specific tasks that must be accomplished in each time step (hereafter referred to as a multigrid cycle) in order to incorporate a multigrid convergence accelerator. The process begins by obtaining a new finest-grid solution based on the predictor-corrector scheme presented above; this is Step 1 of the multigrid sequence as presented in Chapter 4. In Step 2 of the multigrid sequence, information on the finer (in the initial step, the finest) grid must be transferred to a second grid whose resolution is one-half that of the finer. This transfer mechanism is called restriction and is accomplished usually by simple point-to-point transfer (injection) or interpolation/averaging. The particular numerical scheme presented herein makes use of a different, but straightforward definition of the restriction operator I'^{k+1}_{k} (given in Equation 4.45) as explained below.

5.4.1 Flow Field Restriction Operator

Figure 5.3 illustrates the simplicity of this restriction operator for the fluxes. As shown, fluxes are defined on the faces of each finer

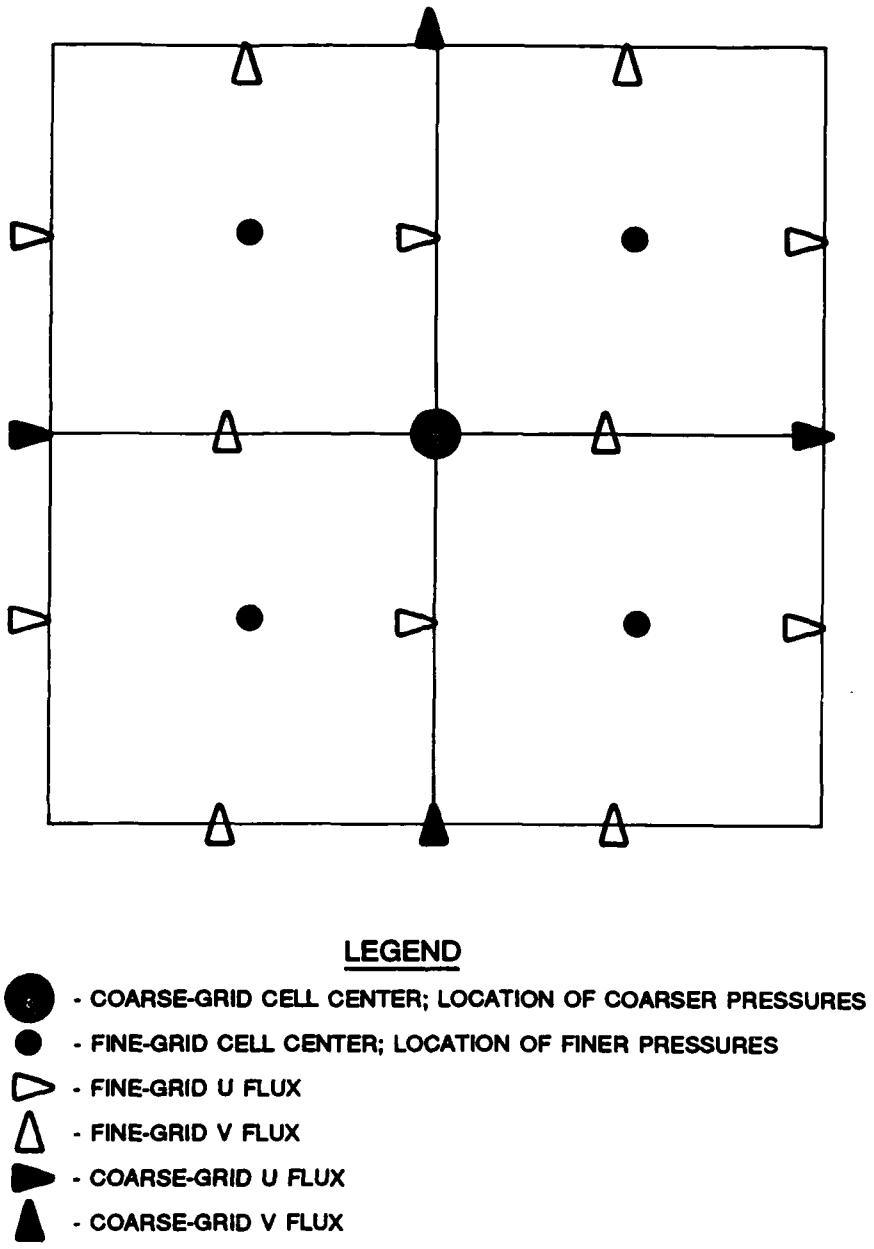


Figure 5.3. Restriction methodology for fluxes

grid cell. When the grid is coarsened to one-half its previous resolution, the coarse fluxes (U_c and V_c) are located at the midpoints of the coarse cell faces. However, these coarse cell faces correspond to the coupling through simple addition of two finer-grid cell faces. Thus, the flux crossing a coarse cell face is nothing more than the summation of the two finer-grid fluxes located on the finer-grid cell faces that

are integrated to make up the coarser cell face. The restriction operator is, then, a simple integration operator for the transfer of the fluxes from finer to coarser grids. In more mathematical terms, this integration method can be thought of as one based on the Gauss Divergence Theorem (Kreyszig, 1979). The integration procedure as outlined is equivalent to computing the surface integral along the faces of the coarse-grid control volume of the velocity component normal to each of these faces times the differential area each acts upon (this product being, by definition, the finer-grid fluxes). This surface integral, based on Gauss' theorem, is exactly equivalent to the volume integral over the coarse control volume of the divergence of the velocity field for each finer-grid control volume times the differential volume of each finer-grid cell. Thus, the restriction operator used herein for the fluxes conserves mass (and mass violations) on the coarser grid exactly the same as on the finer grid. The transfer of flux information from a given level of coarse grid to an even coarser grid is also accomplished in this same way.

The restriction of pressure information is accomplished in a way analogous to that for the fluxes. As shown in Figure 5.3, each coarser-grid cell is made up of four finer-grid cells. At the centers of each of these finer cells resides pressure information. In most multigrid applications, this pressure information would be integrated in a fashion equivalent to the arithmetic averaging of the four finer-grid pressure values, and this value would be restricted to the coarser grid. However, it is not the pressures themselves which are of greatest importance to an incompressible flow calculation; rather, the pressure gradient, and its contribution to mass conservation, is of primary

concern. In an effort to accurately transfer the effects of the finer-grid pressure gradient to the coarse grid, the third terms on the right-hand sides of Equations 5.27 and 5.28 divided by the finer-grid time step are computed. These respective terms represent the contribution of the pressure gradient per unit time to each of the fluxes. Thus, these terms may be thought of as the change per unit time in each of the flux terms due to pressure alone. Each of these pressure terms is centered at the same locations as the fluxes they impact; thus, the pressure gradient is restricted in exactly the same manner as the fluxes.

A third flow component that is restricted from the finest grid to all subsequent coarse grids is the additional shear force generated along no-slip boundaries. The computation of this shear force on the finest grid is discussed in Appendix B for generalized curvilinear coordinates. Simply stated for cartesian coordinates, this shear force is the product of the first derivative of velocity with respect to the direction normal to a given boundary, absolute viscosity, and a differential unit area along the given boundary (i.e., the length of the side of a given cell coexistent with a particular boundary). The forces acting on the boundary cells on the finest grid are integrated in the same manner as the fluxes. This integration occurs only along the no-slip boundaries because these particular shear forces arise from the viscous terms along no-slip boundaries. The remaining components of the viscous terms (i.e., those arising from the treatment of all cells as slip) are computed directly from the given velocity field on any grid as discussed in Appendix B. The integration and restriction of these shear forces from the finest grid to all subsequent coarse grids is employed in lieu of computing these forces directly from the coarse-grid velocity

field for two reasons: (a) these shear forces may be thought of as external body forces acting on the flow domain. They are, however, generated in the direct vicinity of the no-slip boundaries, their computation being a function of velocity derivatives on said boundaries. Computation of these forces directly from coarse grid information results in the calculation of boundary velocity derivatives from field information ever-distant from the boundary itself. This could introduce shear forces out of proportion to the velocity field near the boundary; and (b) experimental evidence showed very clearly that the convergence acceleration obtained by restricting these shear forces from the finest grid to all coarser grids is superior to that obtained when the shear forces are computed directly from the coarse-grid velocity field. In some cases, convergence was completely arrested when the shear forces were computed directly from the coarse-grid velocity field. Apparently this latter computation of the shear forces resulted in the generation of errors on the finest grid, particularly in mass conservation, which the numerical scheme could not dissipate.

5.4.2 Computation of Residual Information

The system of partial differential equations given in Equation 4.5, and solved for on each coarse grid, is not the original system (i.e., Equations 4.5 - 4.7) but a modified system. These modified equations arise because the intent of the coarse-grid computations is to calculate corrections to the finest-grid solution rather than to calculate new solutions themselves, thereby reducing the error associated with the finest-grid solution. The intent of these calculations, then, is the reduction of the residuals as computed by Equation 4.44. As shown in Equation 4.45, the restricted finer-grid residuals make up the second

term on the right-hand side of this equation for each coarser grid. Thus, the calculation and restriction of the finer-grid residuals (Step 3 of the multigrid sequence in Chapter 4) represents an extremely important facet of the multigrid approach.

The residuals for the system of partial differential equations found in Equation 4.44 are calculated as the differences between the known right-hand terms of the equations and their numerical approximations. Recall that the actual system of equations being solved herein is the steady-state analogy to Equations 4.5-4.7 for which the time derivatives of the velocities in the momentum equations are, by definition, zero. Thus, $L'^{k}\underline{u}'^{k}$ in Equation 4.44 represents the numerical approximation to the steady-state version of Equations 4.5-4.7. Rearrangement of Equations 4.5 and 4.6 shows that $L'^{k}\underline{u}'^{k}$ could also be thought of as the numerical approximation to the negative of the time derivatives of respective velocities. Prior to reaching the steady state, these values are obtainable directly from the relaxation of Equation 5.19. Thus, the $L'^{k}\underline{u}'^{k}$ terms for the momentum and continuity equations are calculated as

$$\frac{-\Delta u}{(\Delta t J)} \tag{5.34}$$

$$\frac{-\Delta v}{\Delta t J} \tag{5.35}$$

$$(U_{\xi} + V_{\eta}) \tag{5.36}$$

where J is the Jacobian for each finer-grid cell as defined in Equation 4.10. Each of these terms is obtainable directly from the relaxation

scheme. The actual computation of these terms is explained below.

Given a newly updated solution on a particular grid, this solution is used as the initial input for one final relaxation sweep on that grid. During this sweep, the relaxation scheme computes modifications to the existing flux and pressure fields resulting in a new "interim" solution. Note that Step 7 of the relaxation scheme, which involves adjusting the fluxes by subtracting the gradient of the pressure changes computed from Equation 5.20, is not performed for the residual calculations. Experimental results showed that the deletion of Step 7 during residual calculation provided much improved convergence acceleration as compared with results obtained through inclusion of the step. This was particularly true for nonorthogonal-grid problems of the type presented in Chapter 7. Two reasons for this observation were postulated: (a) the pressure boundary conditions may be inappropriate; and (b) it may be conceptually inappropriate to include Step 7 in the residual calculation. An evaluation of Equations 5.27 and 5.28 revealed that the pressure information "outside" the field is required only in the nonorthogonal-grid cases, such cases being the ones that had convergence difficulties when Step 7 was included in the residual calculation. However, reason (a) appeared somewhat unlikely as the cause of the convergence difficulties because the fine grid-only simulations for the nonorthogonal case converged, albeit slowly, to the correct solution. The reason (b) statement is believed to be correct. The residual calculation actually requires a numerical approximation to the partial differential equations being solved. The computations within Step 7, while part of the relaxation scheme, are not part of the actual equations being solved. Thus, the Step 7 computations may be inappropriate

for inclusion in the residual calculations and are omitted therein.

Changes in the U fluxes (from the relaxation scheme) on the east and west faces of a given cell are averaged to get a U-flux change at the center of the cell; similarly V-flux changes on the north and south faces are averaged as well. Equations 5.15 and 5.16 are then rearranged similar to Equations 5.13 and 5.14 to calculate the changes in u and v at the cell centers. These changes in u and v are then used to solve Equations 5.34 and 5.35. The interim fluxes are then used to solve Equation 5.36 in the form:

$$(U_\xi + V_\eta)_{i,j} = U(i+1,j) - U(i,j) + V(i,j+1) - V(i,j) \qquad (5.37)$$

The interim solution is then discarded and the initial solution restricted to the coarser grid.

Having quantified the calculation of the $L'^k\underset{\sim}{u}'^k$ terms, it remains to calculate the actual residuals and restrict them to the coarser grid. Residual computation on the finest grid is trival because the $\underset{\sim}{f}^k$ terms in Equation 4.44, which represent the right-hand sides of the partial differential equations being relaxed on the given grid, are zero on the finest grid. Thus, the residual terms are merely the negative of the terms in Equations 5.34-5.36. This simplicity does not follow for the coarser grids. As shown in Equation 4.45, the $\underset{\sim}{f}$ terms on all coarse grids are made up of residual and other components which are generally non-zero. In this way, it is the residuals from computations on less-coarse grids which drive the computations on a coarser grid.

Restriction of the residuals is accomplished through volume integration. The residuals are located at the centers of each finer-

grid control volume. Four of these finer control volumes make up a single coarser cell. For each of the residual terms, the values for each of the terms at the centers of the four finer-grid cells are integrated (summed) and stored at the spatial location corresponding to the point where corners of the four finer-grid cells intersect.

5.4.3 Completion of Right-Hand Side Calculations

Having completed restriction of the finer-grid solution to the coarser grid, an initial relaxation sweep on the coarse grid must be made. This sweep is required to evaluate the first term on the right-hand side of Equation 4.45. This term represents the coarse-grid counterpart to the $L'^{k}\underline{u}'^{k}$ terms defined above for the finest grid. The relaxation scheme is employed on the coarse grid using the restricted finer-grid solution as an initial flow-field estimate. The modifications to the finest-grid relaxation scheme are again utilized in these coarse-grid calculations. This relaxation sweep is made while neglecting any effects of the finer-grid residuals. This, in effect, solves the system of equations represented by Equation 4.45 while assuming its right-hand side to be zero. In this way, these calculations allow for the values the terms listed in Equations 5.34-5.36 that would result on the coarse grid in the absence of finer-grid residual effects. Note that the methodology used to compute the $L'^{k}\underline{u}'^{k}$ terms on the finest grid is followed exactly for computation of the L' terms on the coarser grid. Coarse-grid time steps and Jacobians are substituted for the finer-grid values in these calculations.

At the end of the fine and coarse-grid residual calculations, the two terms on the right-hand side of Equation 4.45 are known. An interesting consequence of the residual computation and restriction

operations discussed above is that, given the linearity of the continuity equation, the two "residual" terms cancel each other exactly for this equation. Thus, the residual computations and restriction discussed for the continuity equation are not necessary and, consequently, are not performed. The actual equations to be relaxed further on each coarse grid for the momentum equations look like Equation 5.19 with two additional residual terms appended to the right side. The continuity equation, Equation 5.20, is relaxed in its original form on each coarse grid.

5.4.4 Coarse Grid Relaxation

Following residual computation and restriction, the predictor-corrector (relaxation) scheme is then employed to obtain updated coarser-grid estimates to the fluxes and pressures on the coarse grid. This constitutes the initiation of Step 4 of the multigrid sequence outlined in Chapter 4. This is accomplished in a manner similar to the procedure presented for the finest grid in the seven relaxation steps discussed previously in this chapter. There are two modifications to the finest-grid relaxation sequence, both consequences of the pressure restriction method outlined above. First, the restricted pressure gradient information, when multiplied by the coarse-grid time step and subtracted from the appropriate coarse flux, represents the adjustment of flux listed in Step 5 of the relaxation scheme. Thus, on each of the grids other than the finest, Step 5 of the relaxation scheme is accomplished based on restricted pressure gradient information rather than upon gradient information computed directly upon the coarse grid. Second, given the particular restriction procedure, the coarse pressure values can be thought of as initially being zero at the beginning of a

relaxation sweep. Changes to the initial pressure field are computed on each coarse grid in Step 6 of the relaxation scheme; Step 7 is then performed on each coarse grid just as on the finest grid. However, at the end of the predictor and corrector steps, changes to the initially zero-valued coarse-grid pressure field have been computed. Equation 5.33 is then used to compute the change during the entire relaxation sweep in the pressure field (assuming the initial pressure field, p^n, to be zero) rather than the new coarse pressure field itself. The gradient of these coarse pressure changes is then computed and added to the gradient information restricted from the finer grid. In this manner, the impact of the changing pressure gradient upon the mass fluxes is correctly transferred from grid to grid. For each coarse grid it is assumed that $\Delta\xi$ and $\Delta\eta$ are still identical to unity for computational purposes. This assumption dictates a recomputation of the metrics and Jacobians for each coarse grid. These values are computed in an analogous manner to their finest-grid counterparts as given in Equations 5.4-5.11, except that the index steps are increased to reflect the coarser grid.

The new coarse-grid solution is then transferred via restriction to a still coarser grid (Step 5 of the multigrid sequence) and Steps 2-4 of the multigrid sequence are again employed. This continues until the coarsest grid N, which is selected by the user, has undergone relaxation. One or several relaxation sweeps can be used on each grid, including the finest, prior to restriction and movement to a coarser grid. However, when relaxation is completed on the coarsest grid, the coarse corrections must be transferred back to the next finer grid through the prolongation process (Step 6 of the multigrid sequence).

This process is discussed below.

5.4.5 Prolongation Operator

A number of prolongation operators, I'^{h}_{h+1} in Equation 4.46, have been discussed in the literature. The numerical scheme used herein results in a much simplified prolongation operator for the flux correction transfers. Recall that each coarser-grid updated flux has an analogous original coarse flux as restricted from the finer grid. Subtraction of these two values results in a coarse-grid correction which must be prolonged to the finer grid. Recall, however, that each coarser flux actually consists of the integration of two finer-grid fluxes. Thus, the coarser-grid correction can be transferred to the finer-grid fluxes through a rule which apportions this correction based on the percentage of the coarser cell face contributed by the faces of each of the two finer cells. This, in effect, provides for the transfer of the greater correction to the finer-grid flux that represents flow across the larger finer-grid cell face.

The pressure prolongation operator is somewhat more computationally demanding than that for the fluxes. At the center of each coarser-grid cell, which is made up of four finer-grid cells, a pressure correction is obtained. This correction represents a change to the original coarse pressure over the entire coarser control volume. However, the pressure correction in each coarser control volume influences not only the finer cells within, but the finer cells sharing a common boundary point with it as well. As shown in Figure 5.4, the four finer-grid pressures are influenced directly by the four coarser pressures. Prolongation of the coarser-grid pressure corrections was therefore affected through a bilinear interpolation scheme that used the four coarser values as

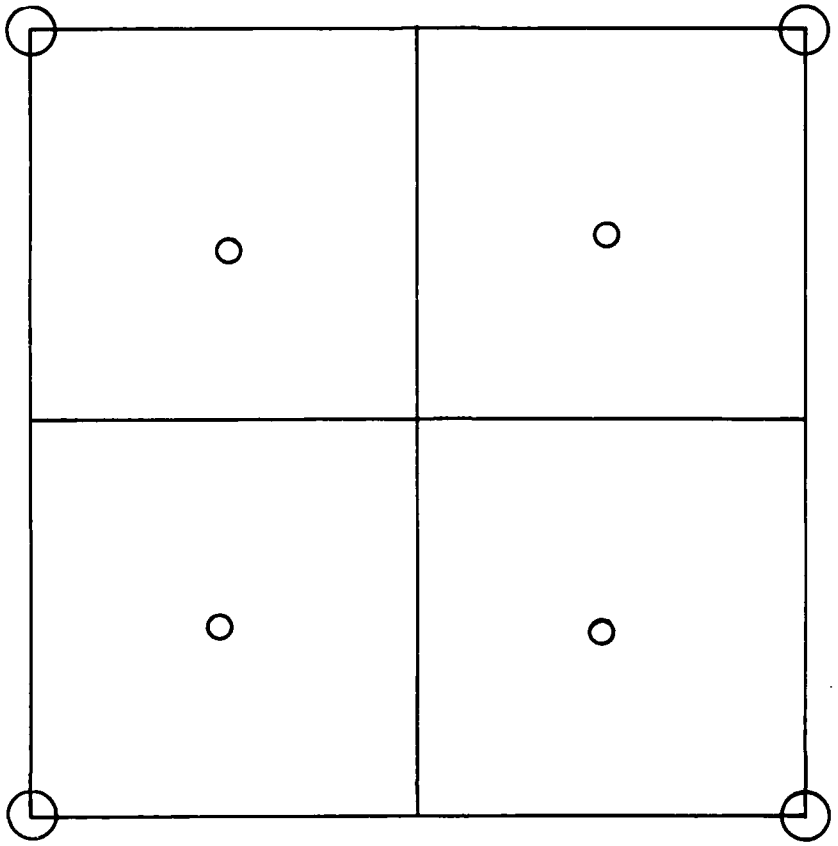


Figure 5.4. Relationships for pressure prolongation

knowns. This scheme is presented in Appendix C.

As each of the flux and pressure corrections is prolonged from coarser to finer grids (Step 7 of the multigrid sequence), it is interpolated to a spatial location already having some flow-field information. If the prolongation step is applied from any coarse grid to a finer grid that is not yet the finest, then the prolonged correction is added to the correction already resident on that grid. If the receiving grid is the finest, the prolonged corrections represent the summation of the corrections computed on the coarse grids. These

corrections are added as shown in Equation 4.46 to the existing solution on the finest grid, thereby creating a new solution at the new time step, n+1. At this point, one entire multigrid cycle has been completed. This cyclic process continues until convergence is reached or a maximum number of iterations is completed (Step 8 of the multigrid sequence).

Given this overview of the numerical solution procedure, it is yet necessary for several additional solution pieces to be discussed. Criteria for defining the maximum number of coarse grids allowable, as well as boundary and initial condition development and time step computation, are discussed in the next chapter.

CHAPTER 6

ADDITIONAL NUMERICAL CRITERIA

6.1 INTRODUCTION

A very important aspect of the numerical solution of the incompressible Navier-Stokes equations for approach flows to hydraulic structures involves the assignment of appropriate initial and boundary conditions. This chapter discusses the formulation and incorporation of these conditions. In addition, the criteria that must be conformed to for stable numerical solutions are presented. Finally, the criteria used to determine the maximum number of coarse grids allowable is explained.

6.2 BOUNDARY CONDITION IMPLEMENTATION

The numerical treatment of boundary conditions is just as important as the discretization of the governing equations presented in the previous chapter. While the governing equations serve to dictate the transmission of information through space and time, the transport is constrained by the boundaries. Regardless of the accuracy of the numerical scheme, poor development and implementation of the boundary conditions will result in poor results.

Bernard (1988) gives a categorization of boundary conditions that will be adopted herein:

a. Definite Boundary Conditions: those that arise from some known (or assumed) physical constraint.

b. Indefinite Boundary Conditions: those that arise only because the computational flow field is smaller than the physical flow field. Boundary conditions for pressure generally fall into the former category, while those for velocity and flux may fall into either according to the problem being simulated.

To begin this discussion, the cells in the vicinity of a given cell (labeled cc) are shown in Figure 6.1. Suppose that the east cell face

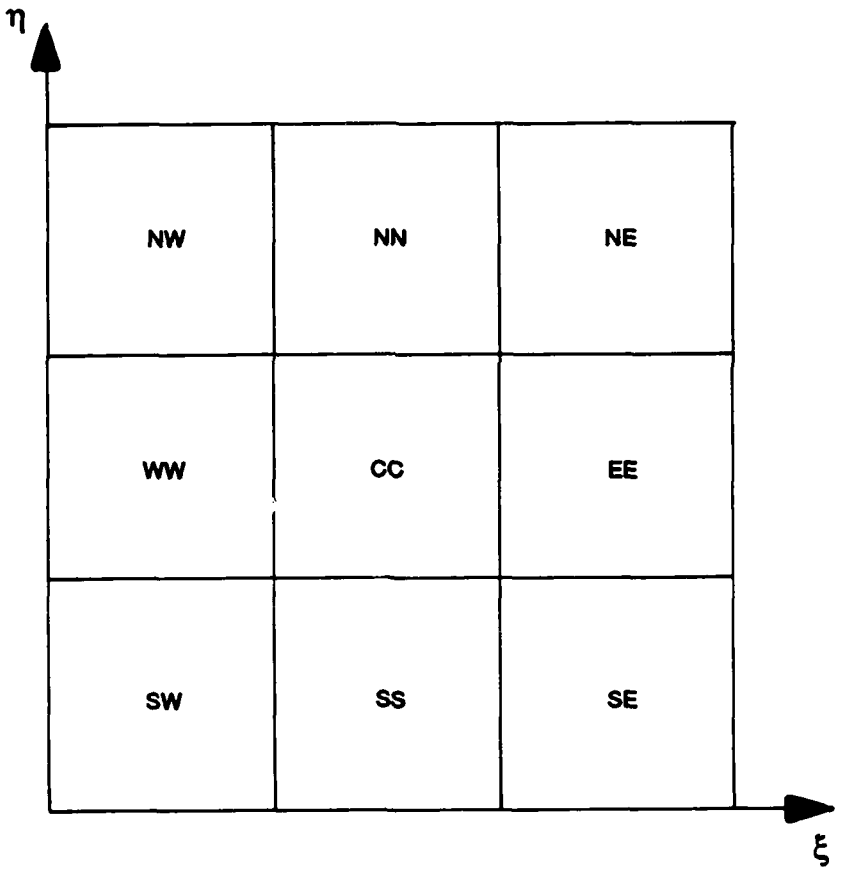


Figure 6.1. Labeling system for cell-centered quantities in neighboring grid cells

of cell (cc) coincides with the outlet boundary, but the north face is in the field, as shown in Figure 6.2. The η -derivative of pressure on the north face could then be computed as

$$p_{\eta}^{n} = p_{nn} - p_{cc} \qquad (6.1)$$

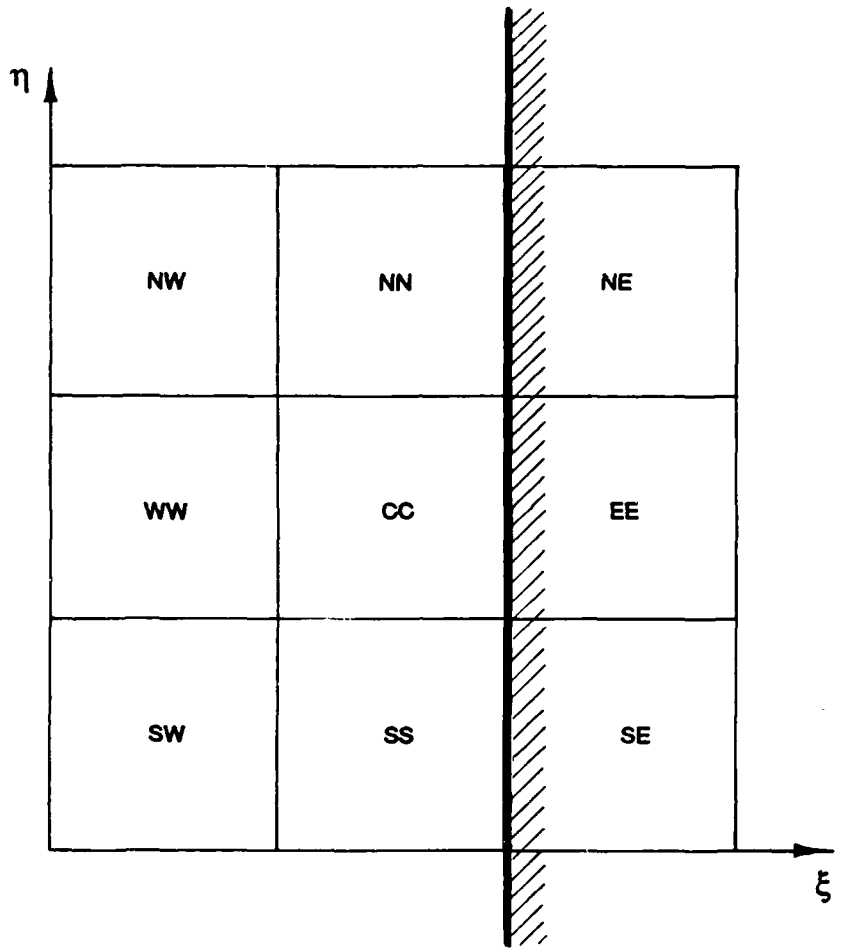


Figure 6.2. Schematic of a given cell (cc) whose east face is coincident with a physical boundary

where the indices represent the cells shown in Figure 6.1. This computation is straightforward and requires information only in the field. However, a problem arises when the ξ -derivative of pressure on the north face is computed. One way to calculate this derivative is

$$p_{\xi}^{n} = 1/4\ (p_{ne} - p_{nw} + p_{ee} - p_{ww}) \qquad (6.2)$$

which requires information "outside" the outlet boundary. This computation cannot be made until the exterior pressures are provided. Bernard (1987) suggests that a different way to compute this derivative is

$$p_{\xi}^{n} = 1/2\ (p_{nn} - p_{nw} + p_{cc} - p_{ww}) \tag{6.3}$$

which he shows to provide superior results to Equations 6.2 and other formulations for both orthogonal and highly nonorthogonal grids. Equation 6.3 is seen to be equivalent to Equation 6.2 if the following definitions are used:

$$p_{ne} = 2p_{nn} - p_{nw} \tag{6.4}$$

$$p_{ee} = 2p_{cc} - p_{ww} \tag{6.5}$$

Equations 6.4 and 6.5 are approximately equivalent to a numerical representation of the second normal derivative of the pressure being equal to zero one the boundary. This boundary formulation is employed for all pressure boundary conditions. Analogous expressions for all boundaries are employed. The pressure values outside the boundaries would not be needed if the grid were orthogonal; however, few curvilinear grids are completely orthogonal.

For cells adjacent to no-slip boundaries, the normal component of the momentum flux through the boundary is computed based on the fact that both velocity components on the boundary are zero. This condition can be implemented by requiring that the velocity outside the boundary be the negative of the corresponding field velocity just inside the boundary. Thus, if the south boundary of the flow field where designated no-slip, then

$$u(i,j-1) = -u(i,j) \tag{6.6}$$

$$v(i,j-1) = -v(i,j) \tag{6.7}$$

where the cell (i,j) is the field cell just inside the south boundary. However, evaluation of this condition revealed that the approach produces an incorrect shear stress on the no-slip boundary when compared with the analytical solution for Couette flow with a pressure gradient (see Chapter 8). A second-order boundary condition for specifically velocity on no-slip boundaries was developed as shown below for a south boundary

$$u(i,j-1) = \frac{(-6u(i,j) + u(i,j+1))}{3} \tag{6.8}$$

This condition assumes the south boundary has zero u and v velocities. However, model tests cases having no-slip top boundaries with non-zero u velocities (i.e., the Couette flow in the absence of a pressure gradient case) are also simulated. Specific second-order boundary conditions for these cases, along with the development of Equation 6.8, are given in Appendix B.

This boundary information is required by the predictor-corrector scheme of the solution of the transport equations for momentum. It should be noted that the cells "outside" the boundary are not actual cells in the flow field and the information within them (pressure and velocities) is not physical. In fact, this boundary information is, in every case, projected from a combination of information derived from within the flow field. Still, for clarity, this information is referred to as being outside the field.

All of the boundary conditions discussed previously are definite because they impose known or assumed constraints on the flow variables. This is true as well for solid boundaries which have zero flux and for boundaries for which the fluxes are specified functions of time. All boundaries are definite with regard to flux in this report.

These same boundary conditions were utilized on all grids of the multigrid sweep with appropriate modifications to reflect the increased coarseness of the multiple grids.

6.3 INITIAL CONDITION SPECIFICATION

The specification of initial conditions for the simulations presented herein, in contrast to the boundary conditions above, was very simple. For all simulations the initial pressure field was set to zero. The velocity field was initially set to a value which provided either no flow or uniform flow throughout. The velocities "outside" the flow field were initially set based on the desired value of the initial fluxes on the boundaries. Thus, the initial boundary fluxes are computed from the velocities just beyond their respective boundaries. All fluxes on no-slip boundaries were set to zero. The boundary conditions were then employed to ensure conformance by the initial flow field.

6.4 STABILITY CRITERIA

The numerical solution scheme utilized herein is an explicit one whose time step must be constrained to promote computational stablity. Although no CFL (Courant, et al. (1929)) limit on the time step can be computed directly for the incompressible solver discussed in Chapters 4 and 5, a heuristic computation can be made. Note that the change in any function f in a time step due to velocity-related processes (advection

and diffusion) should not be larger than the function's value at the beginning of the time step. This, in turn, acts to keep the sign of the function f from changing wildly in a time step, thereby promoting stability. This can be expressed mathematically as

$$\frac{\Delta f}{f} = \Delta t A \leq 1 \tag{6.9}$$

where Δt is the time step of interest and A represents the coefficients of the advective and diffusive terms related to the function f. Thus, from Equation 6.9,

$$\Delta t \leq \frac{1}{|A|} \tag{6.10}$$

where the absolute value is taken to ensure a positive time step. For the nonconservative form of the governing incompressible equations in generalized curvilinear coordinates (Equations 4.5 and 4.6). Equation 6.10 transforms into the constraint that

$$\Delta t \leq \frac{1}{[2J(|U| + |V| + \nu J\{y_\eta^2 + x_\eta^2 + x_\xi^2 + y_\xi^2\})]} \tag{6.11}$$

where the values in the [ ] brackets are those which multiply the velocities u and v in the nonconservative form of these equations. Because this formulation fails to account directly for non-linearities in the governing equations, a factor of safety is multiplied by the time step computed from Equation 6.11. This factor is always between 0 and 1, with the usual value being 0.9. This stability requirement is

computed at each point in the field upon initialization of the flow field only. The most constraining (smallest) time step computed in the field is then used as the global time step for all iterations and field locations for that grid. No local time stepping is used in these calculations.

A second stability criterion for these calculations is required due to the use of pseudo-compressibility. In Chapter 4 the β coefficient, representing the pseudo-sound speed in these calculations, is introduced from the original Poisson pressure solution. Equation 4.37 gives the cartesian formulation of β . For stable calculations, β must be less than or equal to this value. Transforming Equation 4.37 into curvilinear coordinates, the stability criteria becomes

$$\beta^2 \leq [J^2(x_\xi^2 + y_\xi^2 + x_\eta^2 + y_\eta^2)\ 2\ \Delta t^2]^{-1} \qquad (6.12)$$

where all the variables are as defined previously. Equation 6.12 is also multiplied by a coefficient between 0 and 1 to provide a stability cushion for the calculation. Both of these coefficients are supplied as input to the code. The limiting values of the time step and β coefficient are recomputed for each of the coarser grids from the initial restriction values imposed on each grid. This is done only once for each coarse grid.

6.5 MAXIMUM NUMBER OF GRIDS ALLOWABLE

The number of grids used in a given simulation is also input by the user. However, care must be taken that this number is compatible with the number of grid points set on the finest grid. The level of resolution on a coarse grid is one-half that of the just-finer grid. Thus,

each coarser-grid cell side would have cell sides encompassing two initially finer-grid cell sides, and so on. This means that, in order for the bookkeeping of spatial locations to be kept straight and for all coarse-grid boundaries to coincide with the finest-grid boundaries, the allowable number of finest-grid cell sides encompassed by each coarser grid cell must be an integer multiple of the number of finest-grid cells along both the ξ and η axis. Expressed mathematically, the maximum number of grids allowable given a certain number of finest-grid points along the x (XPTS) and y (YPTS) axes is determined by the following trial and error method

$$XMOD = MOD\left[\frac{(XPTS - 1)}{2^{NGRIDS-1}}\right] \tag{6.14}$$

$$YMOD = MOD\left[\frac{(YPTS - 1)}{2^{NGRIDS - 1}}\right] \tag{6.15}$$

where NGRIDS is the number of total grids one wishes to utilize; and the term "MOD" represents the mathematical function that returns the modulus of its argument. In order for the NGRIDS value to be allowable, the XMOD and YMOD values must both be very near or equal to zero. This set of operations is tried for several NGRIDS values, the largest satisfying the above criteria being designated maximum. A maximum of 4 grids are generally used in this effort even if the maximum allowable number of grids as computed above is greater than 4. This is done strictly as a matter of convenience.

CHAPTER 7

MASS CONSERVATION CASE STUDY

7.1 BACKGROUND

To illustrate the utility of the algorithm presented herein in minimizing continuity violations, a case study involving mass conservation is next presented. Prior to providing details of this case, several items concerning the presentation format are explained. The contents of the next few paragraphs are applicable not only to this chapter but to the next chapter as well.

7.1.1 Convergence Histories

Plots of convergence histories are presented for each case study simulated. The norms plotted are obtained for the U and V fluxes from the absolute differences between the simulated and analytical solutions. Both maximum-in-the-field and field-averaged values are presented in tabular form for these flux norms. This is done to allow an evaluation of how well the worst flux in the field is converging compared to the field as a whole. Only the field-averaged norms are presented in graphical form for the flux convergence histories.

The pressure norms are computed as the differences in the pressures at consecutive time steps divided by the time step between the calculations. Both field-averaged and maximum-in-the-field values are presented in tabular form.

The norms for the divergence of velocity are computed as the true right-hand side of Equation 4.7 (actually, its two-dimensional analog).

Only the maximum-in-the-field norm is presented in this report since it is the cell-to-cell violation of continuity that contributes to error in incompressible flow computations rather than a field-averaged value.

7.1.2 Speed-Up Factors

Given in tabular form for each case study are speed-up values that represent the relative computer processing time required to reach some level of convergence with no multigrid (one fine grid only) divided by the processing time required to reach the same convergence level with multigrid added. Relative computer processing (or run) time is defined as the time (in computer processor seconds on a CYBER-205) required to complete one multigrid cycle (or, for the fine-grid-only runs, one iteration) multiplied by the number of cycles or iterations needed to drive the convergence norms to some pre-defined tolerance. Note that this tolerance is case-specific and is sometimes chosen based on the norms of the converged fine-grid-only simulation.

Speed-up factor is presented as an index of the relative acceleration of convergence associated with various multigrid scenarios. It is presented because the speed-up factor as computed gives a straightforward evaluation of the computing resources saved by employing the multigrid approach as compared to using a single grid alone. This factor is not simply the ratio of the number of iterations required by the fine-grid-only simulation to meet a specified tolerance and its analogous multigrid counterpart; such would be true if incorporation of the multigrid approach did not increase the number of operations, and thus the computing time, required to complete one "time step" as compared to the fine-grid-only solution. Indeed, multigrid does require an increased number of operations and computing time per iteration or

cycle. Therefore, the speed-up factor considers the actual computing times required to reach a level of convergence for similar multigrid and non-multigrid runs.

7.1.3 Interpretation of Graphical Presentations

For each of the convergence histories presented, data from every 10th iteration or cycle are presented. Symbols are plotted every 50th iteration or cycle. The convergence trends are easily depicted, and the data files housing the information are only 1/10th the size they might otherwise be.

Within the legend on each convergence history there is a symbol, an abbreviation for the norm being plotted (i.e., "DIV" denotes the divergence of velocity), and an abbreviation for the scenario generating that history. The form of this latter abbreviation is "G/F/C" where G denotes the number of total grids used in that simulation; F the number of relaxation sweeps employed in each multigrid cycle on the finest grid; and C the number of relaxation sweeps per grid employed on each coarse grid for each multigrid cycle. The designation 1/0/0 is used to denote the fine-grid-only (or no-multigrid) simulations. While it is realized that the first 0 should, by the above convention, be a 1 (given that the fine-grid-only simulations use 1 relaxation sweep per iteration on the finest grid), this designation is used to more vividly set this simulation apart from those incorporating some level of multigrid.

7.2 CASE STUDY DETAILS

To illustrate the mass conservation properties of the algorithm presented herein, and to initiate examination of the multigrid approach, results from two scenarios for a model case study are presented. The model case involves steady, uniform flow, in the absence of any pressure

gradient. In addition, the effects of the nonlinear advective terms in the equations of motion (Equations 4.5 and 4.6) are considered negligible. The calculations are made for one scenario on an orthogonal 21- by 21-cell grid (Figure 7.1a); and, for the second scenario, on a purposely skewed, nonorthogonal 21- by 21-cell grid as shown in Figure 7.1b. The grid is skewed 45 degrees representing an upper bound on the amount of nonorthogonality this algorithm could be expected to encounter and still produce accurate results. The top and bottom boundaries in each of these scenarios were stationary and slip. A uniform velocity profile with a value of one along the entire inlet and outlet was specified for each scenario as shown in Figure 7.2. These scenarios are analogous to potential flow situations.

The physical domain for each of these simulations is a box with a height-to-length aspect ratio of one for the orthogonal grid, and two for the skewed grid.

The solution procedure is asked to perform a straight-forward activity in each case: given that the fluxes along the inlet and outlet are all constant values (the solid boundaries, of course, have flux values of 0), and that the initial estimates for the fluxes and pressures are universally 0, produce the correct flow fields inside the box. The flux components for the model cases above can be expressed analytically as

$$U = 0.05 \tag{7.1}$$

$$V = 0.00 \tag{7.2}$$

$$\frac{\partial p}{\partial x} = 0.00 \tag{7.3}$$

$$\frac{\partial p}{\partial y} = 0.00 \tag{7.4}$$

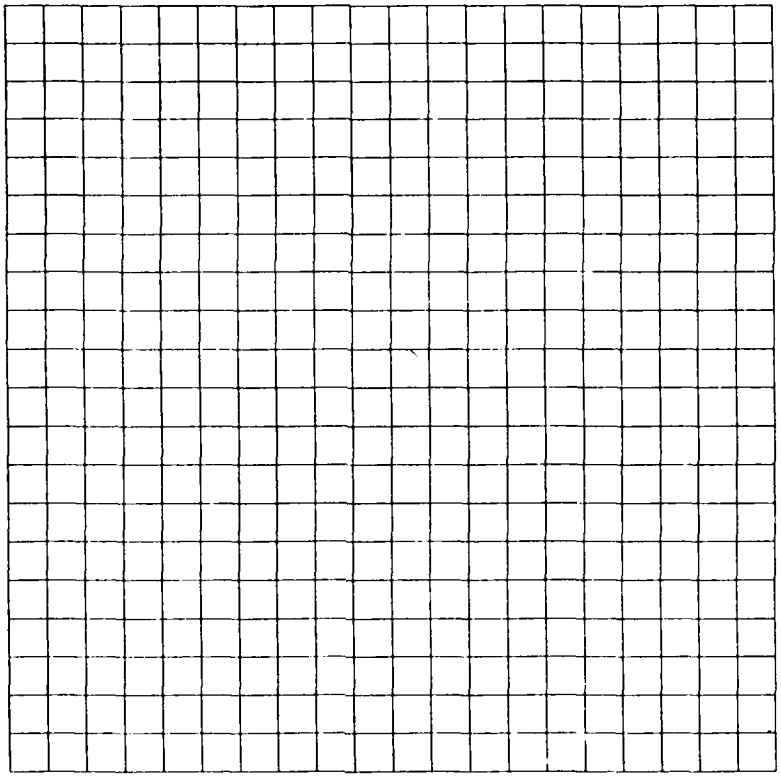


Figure 7.1a. Orthogonal grid

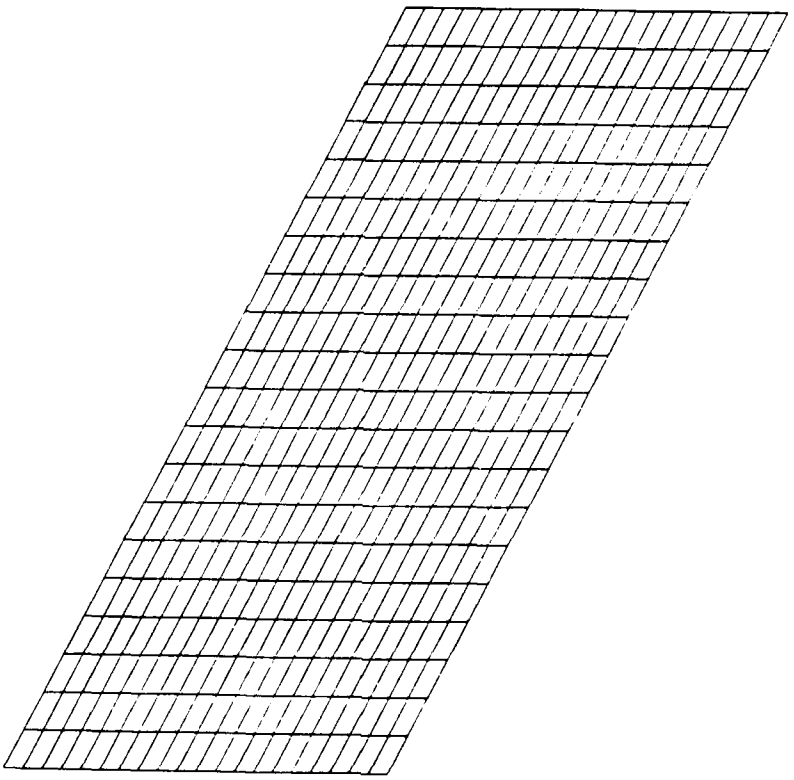


Figure 7.1b. Nonorthogonal grid

Figure 7.1. 21-by-21 orthogonal and nonorthogonal grids

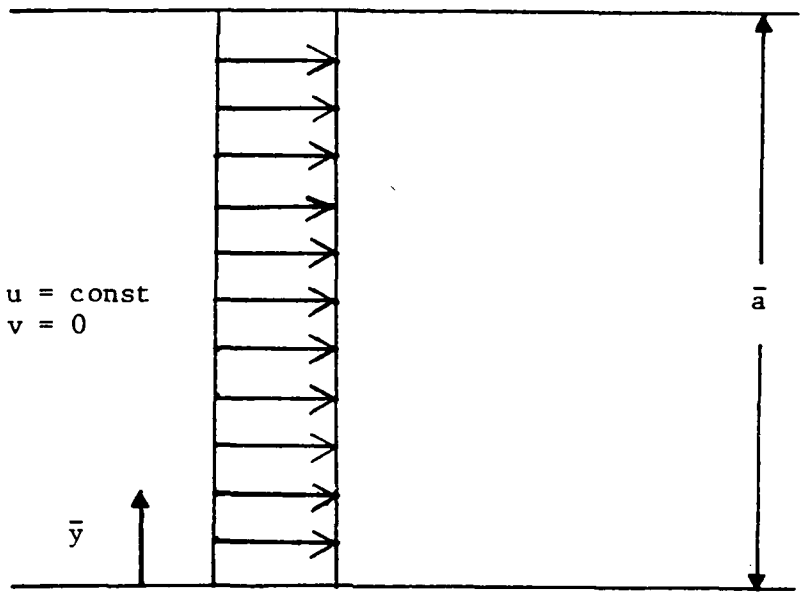


Figure 7.2 *Schematic of mass conservation test setup*

where all variables are as defined previously. The variable p here actually is pressure divided by density.

7.3 RESULTS OF POTENTIAL FLOW SIMULATIONS

7.3.1 Results for Orthogonal, 21-by-21 Grid

Table 7.1 gives a summary of the error norms for each of the fluxes and the pressure as computed after a specified number of iterations or multigrid cycles. The norms for the fluxes are computed as the differences between the computed and analytical values for each run listed. The pressure norms represent the time derivative of pressure divided by density which is introduced by the Chorin scheme. This value should, of course, approach zero at the steady state. Both maximum-within-the-field and field-wide norms (as designated by maximum and average, respectively) are provided for the fluxes and the pressure time

derivative. The addition of multigrid greatly reduced the error norms of the computations as compared to the fine-grid-only solution.

Table 7.1
Summary of Convergence Norms for
Multiple Grid Runs: Potential Flow, 21-by-21
Orthogonal Grid Problem
Convergence Tolerance - 0.0001

		Average			Maximum		
Run	$\nabla \cdot u$	$\partial p/\partial t$	ΔU	ΔV	$\partial p/\partial t$	ΔU	ΔV
1/0/0	2.1e-04	9.2e-16	2.1e-06	1.1e-15	2.3e-10	3.2e-06	4.3e-15
2/1/1	1.3e-05	1.1e-16	1.2e-07	1.1e-15	2.7e-11	1.9e-07	4.2e-15
3/1/1	1.5e-12	1.5e-24	2.4e-15	1.9e-15	1.3e-18	9.8e-15	5.4e-15
2/2/1	6.7e-06	7.4e-18	6.4e-08	1.2e-17	1.8e-11	9.5e-08	5.6e-15
3/2/1	1.5e-12	1.8e-24	2.8e-15	2.1e-15	1.4e-18	9.8e-15	6.4e-15
2/2/2	8.6e-10	1.3e-20	8.3e-12	1.3e-15	3.1e-14	1.2e-11	5.6e-15
2/3/2	9.9e-12	3.7e-20	2.7e-14	1.7e-15	9.6e-16	4.1e-14	7.0e-15
2/3/3	6.3e-13	1.2e-22	2.0e-15	1.9e-15	1.1e-17	8.9e-15	4.9e-15
3/2/2	7.5e-12	8.4e-22	5.4e-15	2.4e-15	7.8e-17	1.9e-14	7.7e-15
3/3/2	5.1e-12	1.9e-20	3.8e-15	2.5e-15	2.1e-16	1.3e-14	7.3e-15
3/3/3	1.6e-11	6.1e-20	8.3e-15	2.2e-15	8.5e-16	4.0e-14	9.2e-15
2/1/2	7.4e-09	8.4e-18	7.0e-11	1.4e-15	2.0e-13	1.1e-10	4.8e-15
3/1/2	5.6e-12	3.8e-22	4.8e-15	2.5e-15	3.4e-17	1.2e-14	1.3e-14
2/2/3	6.0e-13	1.3e-22	2.9e-15	1.8e-15	1.1e-17	8.4e-15	5.2e-15
3/2/3	1.3e-11	3.1e-20	8.0e-15	2.4e-15	3.9e-16	3.2e-14	8.5e-15
2/1/3	7.8e-13	1.3e-22	2.0e-15	1.9e-15	9.3e-18	8.4e-15	4.9e-15
3/1/3	5.8e-11	1.5e-19	2.9e-14	2.0e-14	2.1e-15	7.6e-14	1.4e-13

Convergence histories for the divergence of the velocity field and the average flux norms for each of the number-of-grids/number-of-fine-sweeps/number-of-coarse sweeps combinations simulated are also given in Figures 7.3-7.5. The computed flow fields generally approached the analytical solution quite quickly for the multigrid runs. The scheme produced no false fluxes as evidenced by the constant V-flux norms of approximately zero. Observe that the residuals from the continuity equation were generally the last (compared to the momentum equation residuals) to reach any specified convergence tolerance. Once the continuity violation had been reduced below 0.0001, no discernible difference in the flux field from its analytical solution was observed. Once this level of convergence was reached, the gradient of ıe computed

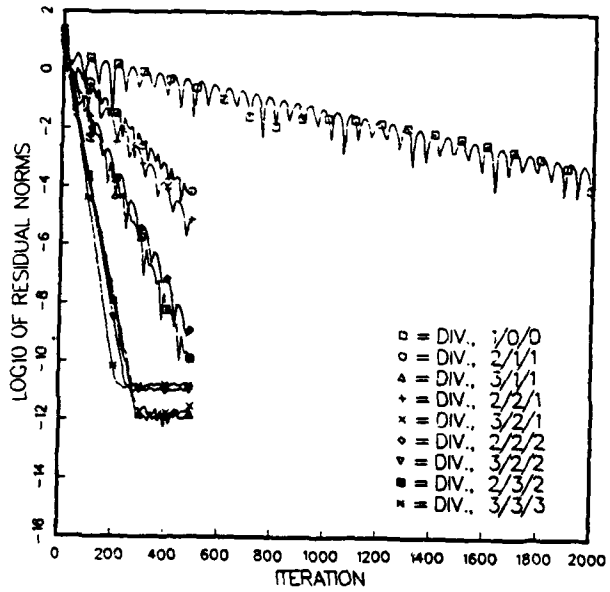


Figure 7.3a. Fine-grid-only and 8 of 16 multigrid simulations

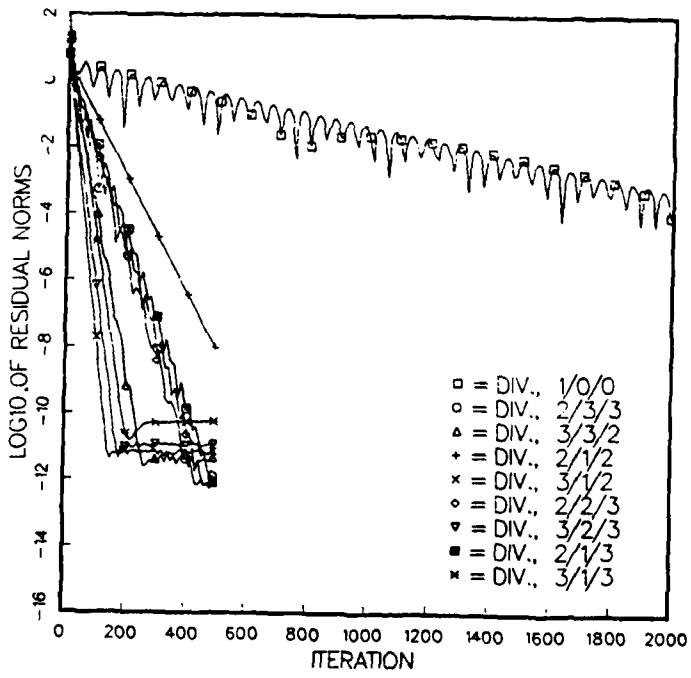


Figure 7.3b. Fine-grid-only and remaining multigrid simulations

Figure 7.3. Convergence history for divergence of velocity, 21-by-21 orthogonal grid, potential flow, multiple grid runs

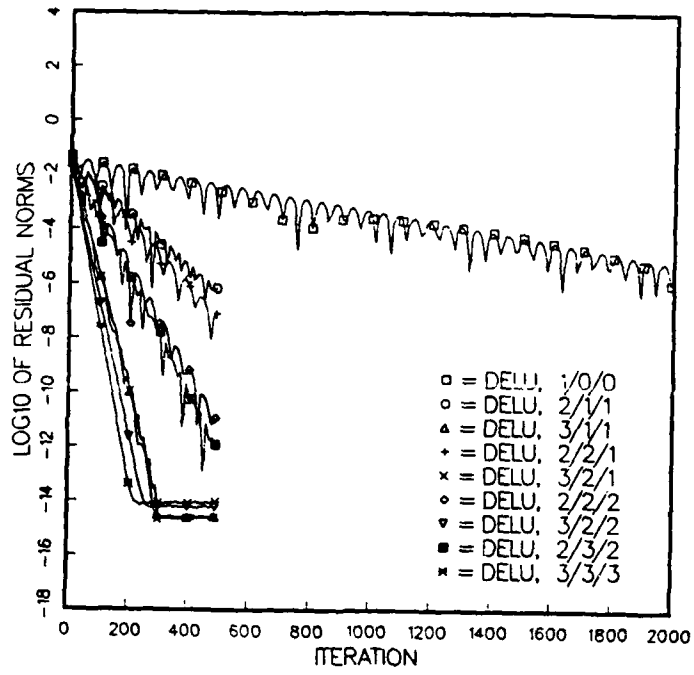


Figure 7.4a. Fine-grid-only and 8 of 16 multigrid simulations

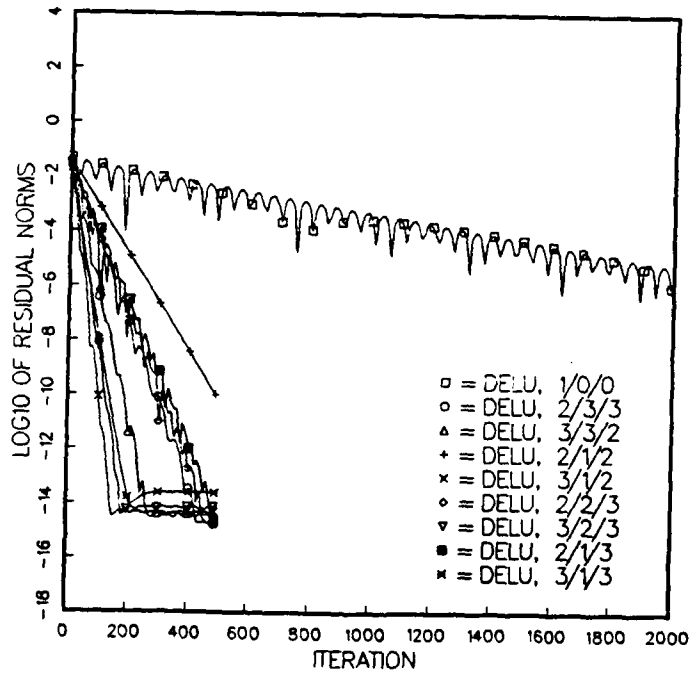


Figure 7.4b. Fine-grid-only and remaining multigrid simulations

Figure 7.4. Convergence history for U flux, 21-by-21 orthogonal grid, potential flow, multiple grid runs

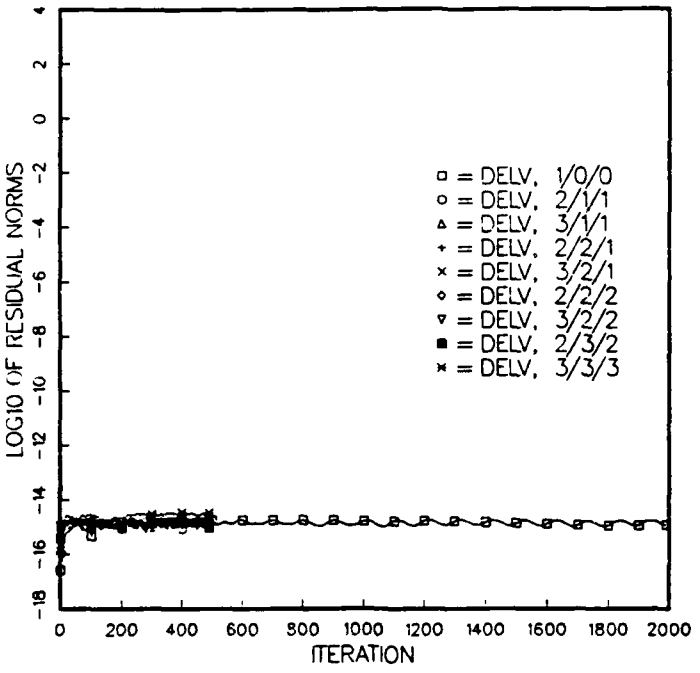


Figure 7.5a. Fine-grid-only and 8 of 16 multigrid simulations

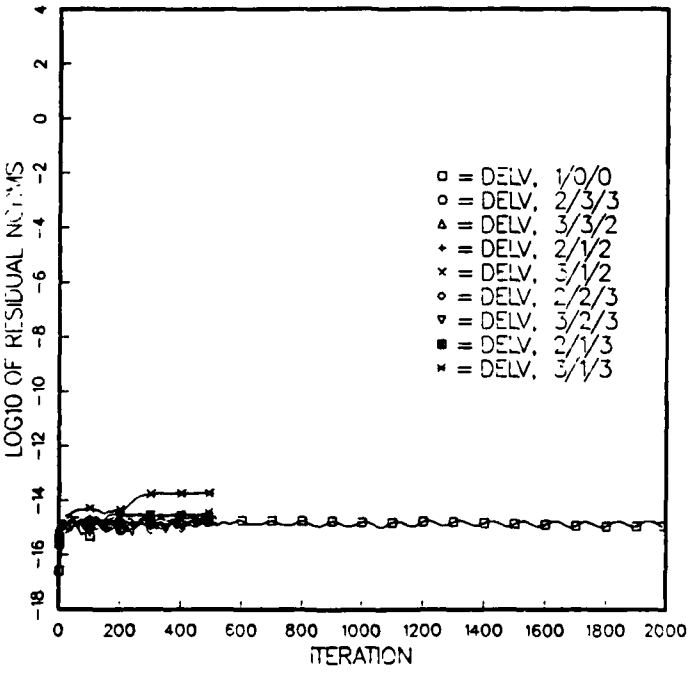


Figure 7.5b. Fine-grid-only and remaining multigrid simulations

Figure 7.5. Convergence history for V flux, 21-by-21 orthogonal grid, potential flow, multiple grid runs

pressure field was approximately zero as well. Therefore, the convergence tolerance for this scenario was set at 0.0001 to permit comparison of the multigrid runs with fine-grid-only final results.

Note that several hundred iterations were required to drive continuity violations to below the assigned convergence tolerance. Had this problem required additional resolution, or the physical domain been spatially larger, the number of iterations required to achieve convergence could have become intractable without the multigrid convergence acceleration.

The fine-grid solution (designated as 1/0/0) was observed to fluctuate as it proceeded to the steady state as shown in Figures 7.3-7.5. These fluctuations are typical of the MacCormack predictor-corrector scheme. Due to the one-sided nature of the differences used in the predictor and corrector, and the changing directions of these differences (four combinations of forward and backward differences that are required to maintain computational stability for incompressible flow), the scheme approaches but may never actually reach the true steady state. However, use of the multigrid approach clearly alleviates much of this problem. The convergence histories for the multigrid runs are much smoother than the fine-grid-only history.

Initial simulations showed that some small amount of dissipation was required by the multigrid scheme to remain stable. This dissipation, which was simulated by computing the viscous contributions for each control volume while assuming it to be a slip cell, was important only in the first few multigrid cycles. Jameson and Yoon (1986) noted that such dissipation is often required for their multigrid scheme in order to smooth oscillations on the fine grid that result from the rapid

modification of the fine-grid solution by coarse-grid corrections. The "kinematic viscosity" used to simulate the results given in this section was between 10^{-4} and 10^{-6}.

Given in Table 7.2 is a comparison of the relative run times to convergence for this flow scenario with no multigrid (1/0/0) and with the multigrid algorithm added. The speed-up factors represent the ratio of the relative run time for the fine-grid-only run divided by that of the given multigrid scenarios. Addition of the multigrid scheme significantly quickened the convergence of the model problem in most cases. In four simulations, however, the multigrid scheme was only mildly more efficient than the fine grid alone in reaching the assigned convergence tolerance. In these cases only one additional level of coarse grid was employed with the result that the overhead of setting up multigrid operations was of the same order of magnitude as the reduction in the number of iterations required to converge as compared to the fine-grid-only run. Thus, care must be exercised in multigrid implementation and utilization.

Table 7.2 shows that the use of multiple rather than single relaxation sweeps on each grid provided for enhanced convergence in some cases on the 21-by-21 grid the three-grid test cases. Brandt (1984) recommends the use of such sweeps. However, while Brandt suggests that 3 may be the optimum number of relaxation sweeps on each grid, 2 such sweeps could be more appropriate for this scheme based on the results presented in Table 7.2.

The results presented above illustrate well a point to note: use as many coarse grids as possible given the resolution of the finest

Table 7.2
Comparison of Convergence Properties for
Multiple Grid Runs: Potential Flow, 21-by-21 Orthogonal Grid
Convergence Tolerance - 0.0001

# Grids	# Fine Sweeps	# Coarse Sweeps	# Iterations	Relative Time to Con. Tolerance	Speed Up Factor
1	N/A	N/A	2000	46.6	1.00
2	1	1	365	27.1	1.72
3	1	1	120	11.1	4.20
2	2	1	300	28.8	1.62
3	2	1	120	13.7	3.40
2	2	2	220	23.7	1.97
3	2	2	120	15.7	2.98
2	3	2	240	31.1	1.50
3	3	2	100	15.2	3.06
2	3	3	150	21.2	2.20
3	3	3	100	16.9	2.76
2	1	2	260	22.3	2.09
3	1	2	60	6.5	7.15
2	2	3	170	20.3	2.30
3	2	3	70	10.3	4.53
2	1	3	185	18.1	2.58
3	1	3	90	11.3	4.14

grid. The maximum number of grids that could be employed for this scenario was 3 for reasons explained in Chapter 6.

A final point to note concerns the stability of the global scheme for certain multigrid simulations. Efforts were made in all cases to run as close to the stability limit as possible (i.e., to within 90 percent of the limit). This limit, for this case study, is based on the maximum β value defined by Equation 6.12. A time step value was first computed using Equation 6.11; however, any time step of finite size would have sufficed due to the absence of major advective or viscous terms in the calculations. Given this time step and the grid geometry, β was computed from Equation 6.12. However, for four multigrid simulations (3/1/3; 3/2/3; 3/3/3; 3/3/2) this value had to be reduced by a factor of from 2 to 4 in order to obtain improved

convergence. This was necessitated by instabilities generated during the start-up of the solution. The instabilities occurred in the first 50 multigrid cycles during which, due to the initial conditions imposed, the continuity violations were very large. It is possible that if more conducive initial conditions were used these instabilities might be mitigated. It is also interesting to note that these instabilities occurred generally for the three-grid/three-sweeps-per-coarse grid simulations, those being the simulations which, by doing the most work on the coarse grids, would return the biggest changes to the fluxes and pressures in the first several cycles.

7.3.2 Results for the Nonorthogonal, 21-by-21 Grid

Results of the 21-by-21 nonorthogonal (skewed) grid simulations are given in Tables 7.3 and 7.4; convergence histories for this scenario for the divergence of velocity and the fluxes are given in Figures 7.6-7.8. The angle of skewness simulated was 45 degrees. A kinematic viscosity coefficient of 0.0001 was used for all the simulations on the skewed grid. Convergence was defined as having been reached when the maximum divergence was less than 0.001. This tolerance was chosen based on the divergence norm for the fine grid-only solution at approximately 5,000 iterations, and the observation that, at this convergence level, no discernible difference in the computed and analytical fluxes was observed. At this convergence level, the computed pressure gradient was approaching zero (i.e., less than 0.0001).

The fine-grid-only and the multigrid simulations did, except for one simulation (2/1/1) that did not reach convergence, produce accurate results. As shown in Table 7.3, the divergence and flux norms were generally from quite to exceptionally small. This result adds validity

Table 7.3
Summary of Convergence Norms for Multiple Grid Runs:
Potential Flow, 21-by-21 Nonorthogonal Grid
Convergence Tolerance - 0.001

		Average			Maximum		
Run	$\nabla \cdot u$	$\partial p/\partial t$	ΔU	ΔV	$\partial p/\partial t$	ΔU	ΔV
1/0/0	7.5e-04	4.4e-13	3.1e-06	5.6e-06	8.1e-09	8.7e-06	9.6e-06
2/1/1	6.1e-03	5.7e-10	2.3e-05	4.0e-05	1.2e-06	5.5e-05	6.5e-05
3/1/1	4.5e-10	5.8e-17	1.3e-12	2.4e-12	1.2e-13	3.6e-12	4.1e-12
2/2/1	1.9e-03	2.9e-10	7.5e-06	1.3e-05	5.9e-07	1.9e-05	2.1e-05
3/2/1	1.3e-08	3.5e-16	5.1e-12	5.3e-11	2.8e-12	2.0e-11	2.6e-11
2/2/2	7.9e-06	1.3e-12	1.7e-08	2.9e-08	2.6e-09	5.2e-08	5.0e-08
2/3/2	7.3e-06	1.8e-12	2.8e-08	4.8e-08	3.6e-09	6.9e-08	7.8e-08
2/3/3	7.0e-08	2.1e-14	2.6e-10	4.5e-10	4.3e-11	6.8e-10	7.4e-10
3/2/2	9.8e-12	8.6e-19	3.3e-14	5.3e-14	2.0e-15	8.6e-14	9.6e-14
3/3/2	2.6e-09	1.2e-16	7.7e-12	1.4e-11	2.4e-13	2.4e-11	2.5e-11
3/3/3	3.4e-12	1.0e-19	6.9e-15	9.5e-15	2.3e-16	2.0e-14	2.7e-14
2/1/2	4.3e-05	1.6e-12	2.6e-08	4.3e-08	3.4e-09	1.2e-07	9.3e-08
3/1/2	4.0e-12	1.6e-19	3.7e-15	5.1e-15	5.5e-15	1.6e-14	1.7e-14
2/2/3	2.0e-07	5.3e-14	7.4e-10	1.3e-09	1.1e-10	2.0e-09	2.1e-09
3/2/3	3.5e-12	5.1e-20	4.4e-15	6.1e-15	2.2e-16	1.3e-14	2.0e-14
2/1/3	1.7e-06	3.2e-13	6.3e-09	1.1e-08	6.6e-10	1.6e-08	1.8e-08
3/1/3	1.1e-04	1.7e-13	9.9e-08	2.5e-08	1.2e-09	2.9e-07	1.0e-07

Table 7.4
Comparison of Convergence Properties for
Multiple Grid Runs: Potential Flow, 21-by-21 Nonorthogonal Grid
Convergence Tolerance - 0.001

# Grids	# Fine Sweeps	# Coarse Sweeps	# Iterations	Relative Time to Con. Tolerance	Speed Up Factor
1	N/A	N/A	4500	104.4	1.00
2	1	1	*	*	*
3	1	1	350	32.3	3.23
2	2	1	950	91.2	1.14
3	2	1	375	42.8	2.44
2	2	2	600	64.8	1.61
3	2	2	300	39.0	2.68
2	3	2	525	67.7	1.54
2	3	3	425	59.9	1.74
3	3	2	350	53.2	1.96
3	3	3	270	45.4	2.30
2	1	2	750	64.4	1.62
3	1	2	160	17.4	5.99
2	2	3	450	53.6	1.95
3	2	3	250	36.8	2.84
2	1	3	550	53.6	1.95
3	1	3	800	100.0	1.04

* This simulation did not reach the convergence tolerance in 1000 iterations.

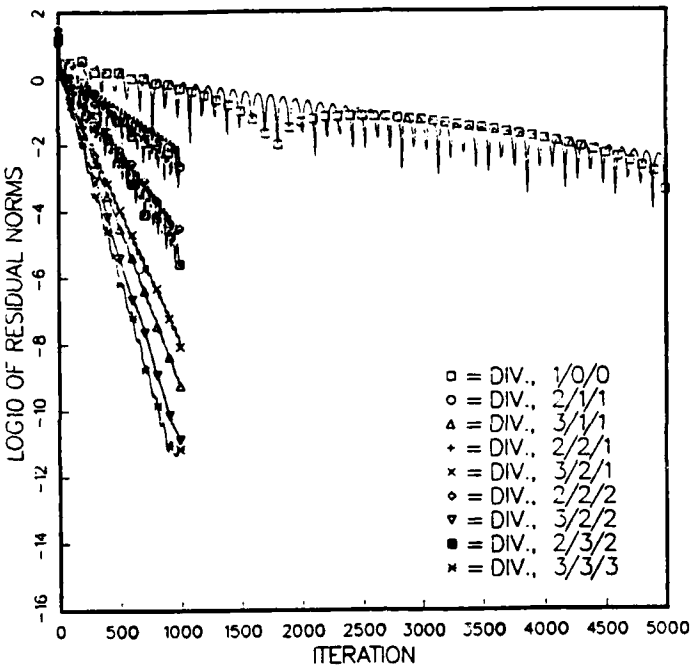


Figure 7.6a. Fine-grid-only and 8 of 16 multigrid simulations

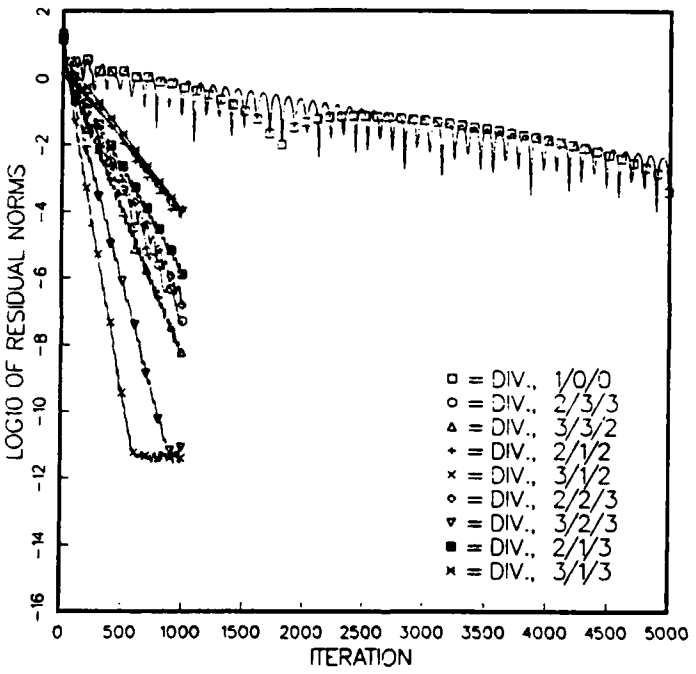


Figure 7.6b. Fine-grid-only and remaining multigrid simulation

Figure 7.6. Convergence history for divergence of velocity, 21-by-21 nonorthogonal grid, potential flow, multiple grid runs

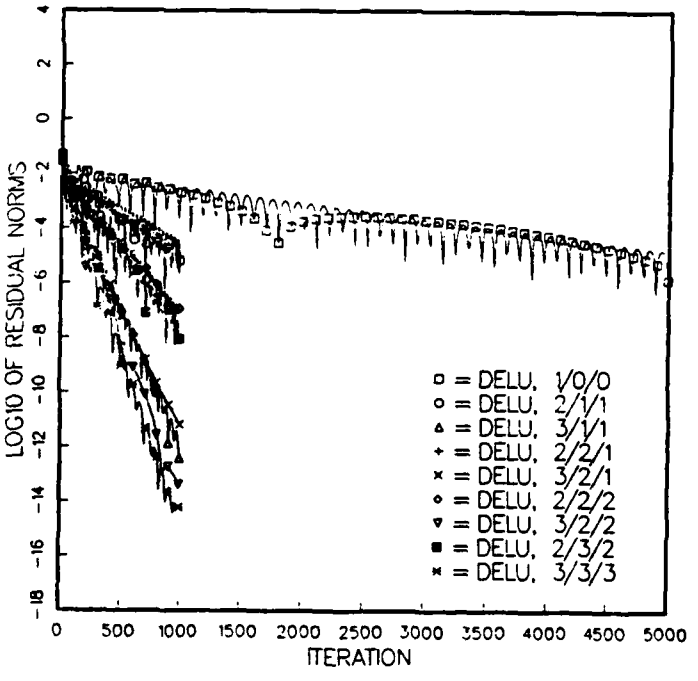


Figure 7.7a. Fine-grid-only and 8 of 16 multigrid simulations

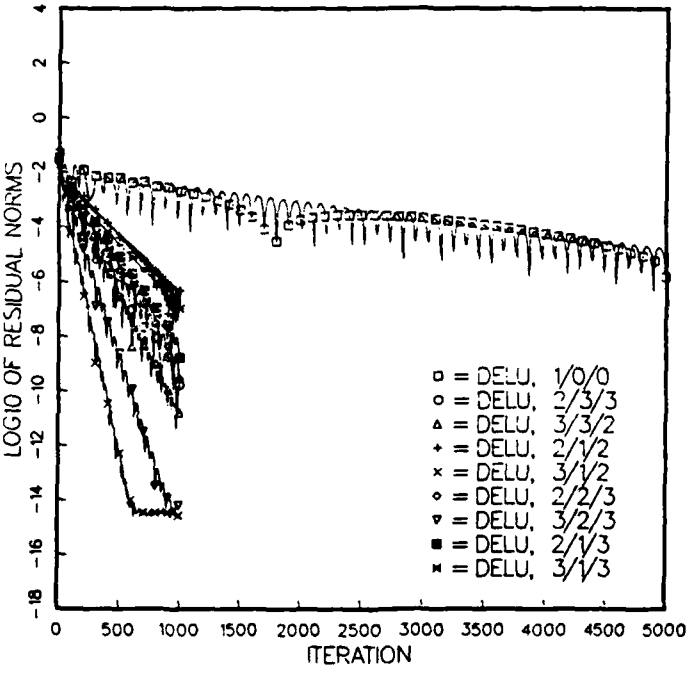


Figure 7.7b. Fine-grid-only and remaining multigrid simulations

Figure 7.7. Convergence history for U flux, 21-by-21 nonorthogonal grid, potential flow, multiple grid runs

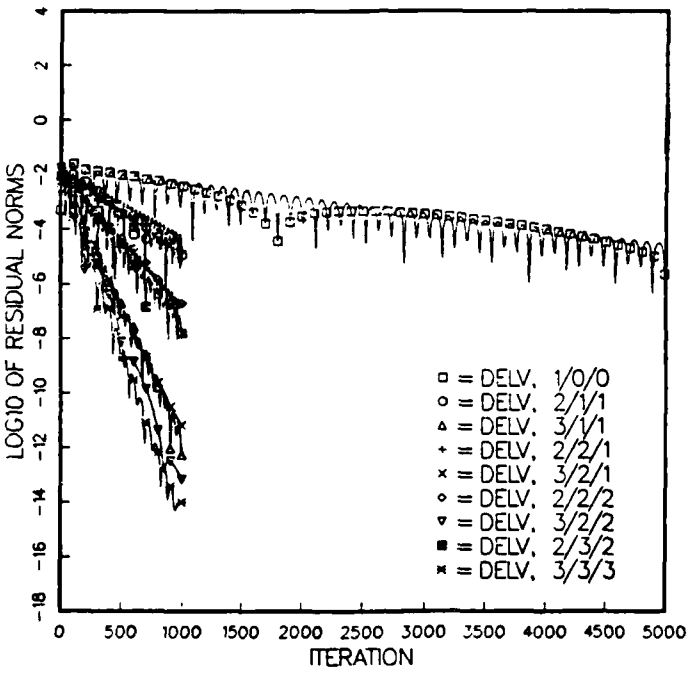


Figure 7.8a. Fine-grid-only and 8 of 16 multigrid simulations

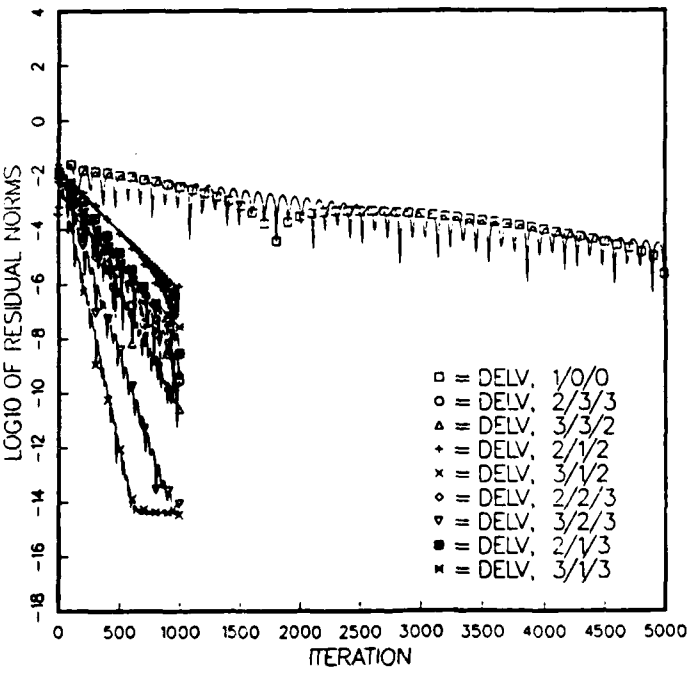


Figure 7.8b. Fine-grid-only and remaining multigrid simulations

Figure 7.8. Convergence history for V flux, 21-by-21 nonorthogonal grid, potential flow, multiple grid runs

to the belief that the algorithm presented in this report is capable of conserving mass in generalized curvilinear space.

The results again indicate that the use of the multigrid approach is warranted. In fact, the use of multigrid may be even more critical for the nonorthogonal case than for the orthogonal case. As illustrated, the nonorthogonal scenario was much slower to converge than the orthogonal scenario. Even after 5,000 iterations, the nonorthogonal 1/0/0 simulation had reached a continuity violation of 7.48E-04. This is contrasted with the orthogonal 1/0/0 run that reached a continuity violation of 2.14E-04 in 2000 iterations. The results also again bear out the need to employ as many coarse grids as possible as shown by the enhanced efficiency of the 3-grid runs over their 2-grid counterparts.

The trends reported for the orthogonal case were generally observed for the nonorthogonal case as well. However, there was some difference in the ordering of the simulations having the greater speed-ups. Still, the same three combinations of three-grid simulations (3/1/2; 3/2/3; 3/1/1) were optimal for both scenarios. Several of the three-grid simulations (3/2/2; 3/3/2; 3/3/3; 3/2/3) had to have adjustments in their β values that were twice that required to ensure stability in the orthogonal runs. This points out that the nonorthogonal simulations require more under-relaxation than their orthogonal cousins to remain stable for the conditions simulated in this case study. Start-up instabilities are also believed to be at the root of these concerns as discussed for the orthogonal case; however, the nonorthogonality definitely further exacerbated this problem.

No effort was made to optimize β values for those runs requiring their adjustment; rather, ever smaller values were simulated until relatively good results were obtained. Thus, the ordering of the three-grid speed-ups could have been slightly different perhaps. The 3/1/3 simulation refused all attempts to adequately accelerate this solution to the correct flow field. As listed in Tables 7.3 and 7.4, results were obtained for this run that were very poor compared to the other three-grid runs. All attempts to modify the β value to improve these results were unsuccessful. The reasons for the inability of this particular simulation of converge quickly for all levels of under-relaxation are not known.

7.4 INITIAL TRENDS FOR FURTHER EVALUATION

At least four initial trends are displayed in the results of the orthogonal and nonorthogonal scenarios. These are: (a) the maximum number of grids possible relative to the level of resolution of the finest grid (as explained in Chapter 6) should be employed at all times to maximize convergence acceleration; (b) some multiple number of relaxation sweeps on both the finest and the coarser grids improves performance of the multigrid scheme. While this actual number is not clearly pointed to by the results, it is apparent that employing fewer (or, perhaps, the same number of) relaxation sweeps on the finest grid than on the coarser grids gives superior performance for this case study. This is supported by the observation that, regardless of specific ordering, the top four simulations in terms of speed-up had fewer fine-grid-relaxation sweeps than the coarser grids did. The cause for this trend is straightforward. The coarser grids have fewer control volumes than the finest grids and, therefore, require fewer operations

per relaxation sweep. Thus, the coarser grids are somewhat more efficient in transmitting information throughout the field than the finest grid alone is; (c) some dampening or artificial dissipation may be required within the multigrid scheme to alleviate problems associated with the interpolation of coarse-grid modifications to the finest grid; and (d) some reduction in the β or time step values may be required in viscous runs to overcome start-up instabilities for some multigrid runs. Unfortunately, the case study presented in this chapter is not rigorous enough to fully support the trends just delineated. Thus, more "real world-like" case studies will be investigated using the trends observed to date as baselines.

CHAPTER 8

VISCOUS FLOW RESULTS

8.1 CASE STUDY DETAILS

To illustrate the utility of the algorithm presented herein for laminar, viscous flow, results from two model case studies are presented. The model cases involve:

a. Couette flow, in the absence of a pressure gradient, at Reynolds numbers of 100 and 400. The calculations are made on orthogonal grids of differing resolution: one a 21- by 21-cell grid (Figure 7.1a); the other 41-by-41 (Figure 8.1). The top boundary in

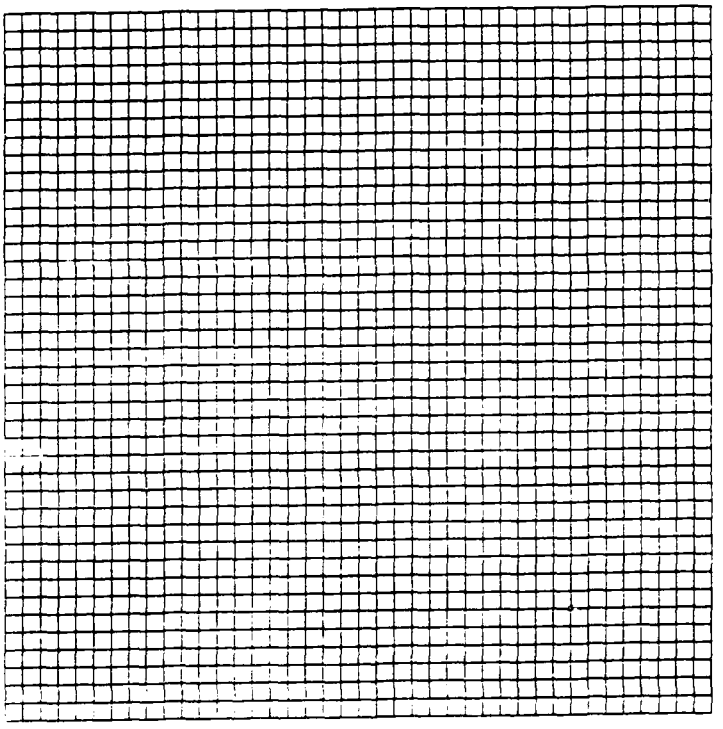


Figure 8.1. 41-by-41 grid for Couette flow simulation

this case is a moving, no-slip boundary, while the bottom boundary is stationary and no-slip. A linearly-changing velocity profile, with a

maximum value of one the top boundary and a minimum value of 0 at the bottom, is specified as constant on the inlet and outlet as shown in Figure 8.2a; and,

b. Couette flow, in the presence of a pressure gradient, at a Reynolds number of 100. The calculations for this case are made on the same grids as listed above. The top and bottom boundaries for this case are stationary and no-slip. A parabolic velocity profile, ranging from zero on the top and bottom boundaries to one at the center of the flow field, is specified as constant on the inlet and outlet as shown in Figure 8.2b. This case will be referred to as stationary viscous channel flow hereafter to differentiate it from scenario (a).

The physical domain for each of these simulations is a square box of unit length.

The solution procedure is asked to perform a straight-forward activity in each case: given that the fluxes along the inlet and outlet were all constant values (the solid boundaries, of course, had flux values of 0), and that the initial estimates for the fluxes and pressures were universally 0, produce the correct flow fields inside the box. Computational results for each of the simulations are compared to the known analytical solutions for the given flow fields in order to assess the efficacy of the former. As given by Streeter and Wylie (1985), the flux components for the model cases above can be expressed analytically as

$$U = \Delta\bar{y}\left[\frac{u_0\bar{y}}{2} - \left(\frac{1}{2\nu}\right)\left(\frac{\partial p}{\partial x}\right)\left(\overline{ay} - \bar{y}^2\right)\right] \qquad (8.1)$$

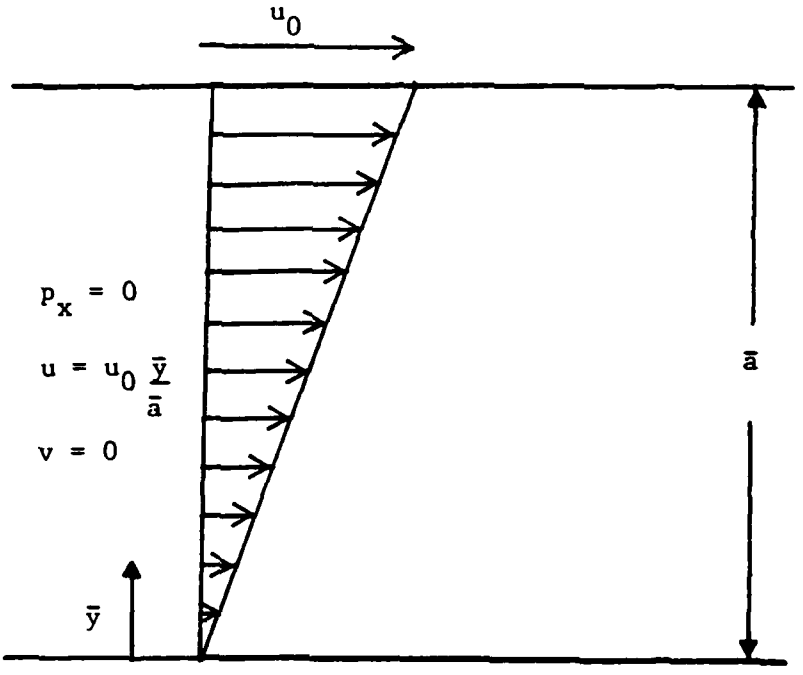


Figure 8.2a. Case without a pressure gradient

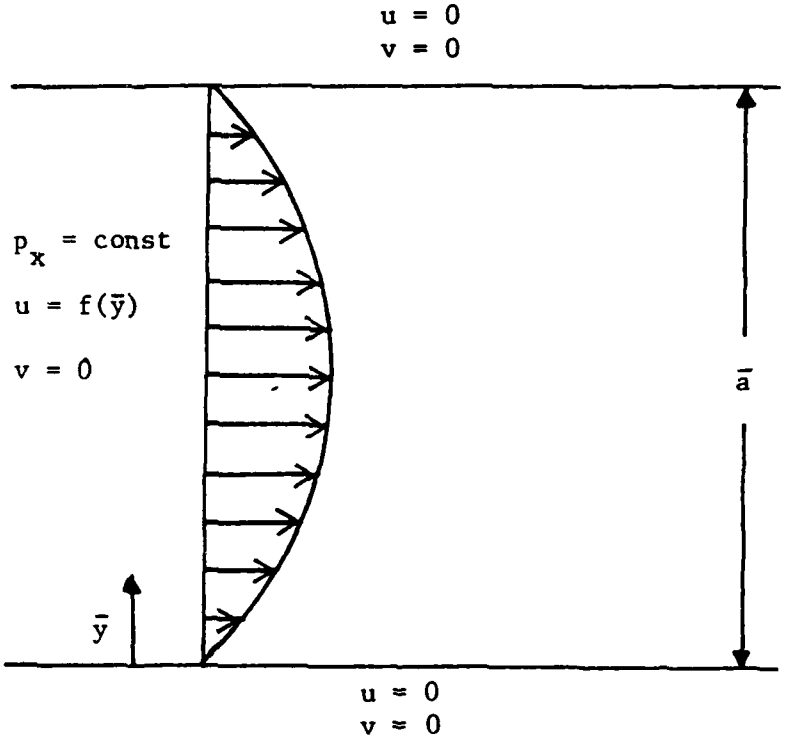


Figure 8.2b. Case with a pressure gradient

Figure 8.2. Schematics for Couette flow test cases

$$V = 0 \tag{8.2}$$

$$\frac{\partial p}{\partial x} = c \tag{8.3}$$

$$\frac{\partial p}{\partial y} = 0 \tag{8.4}$$

where

u_0 = velocity of the top boundary (1 for scenario (a); 0 for scenario (b));

$\bar{a}$ = distance from the top to the bottom boundary;

$\bar{y}$ = distance from bottom boundary to a given point in field;

$\Delta\bar{y}$ = vertical length of a given control volume face;

c = a constant (0 for scenario (a); nonzero for (b))

and all other variables are as defined previously. The variable p here actually is pressure divided by density.

8.2 RESULTS OF COUETTE FLOW SIMULATION

8.2.1 Results for 21-by-21 grid, Re = 100

Table 8.1 summarizes the error norms for each of the fluxes, the divergence of the velocity field, and the pressure as observed after a specified number of iterations or multigrid cycles. The pressure norms represent the time derivative of pressure divided by density that is introduced by the Chorin scheme. This value should, of course, approach zero at the steady state. As presented in the table, the addition of multigrid greatly reduced the error norms of the computations as compared to the fine grid-only solution.

Convergence histories for the divergence of the velocity field and each of the average flux norms for each of the number-of-grids/number-of-fine-sweeps/number-of-coarse sweeps combinations simulated are given

in Figures 8.3-8.5. The computed flow fields generally approached the analytical solution quite quickly for the multigrid runs. The numerical

Table 8.1
Summary of Convergence Norms for
Multiple Grid Runs: Couette Flow, 21-by-21 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 100

		Average			Maximum		
Run	$\nabla \cdot u$	$\partial p/\partial t$	ΔU	ΔV	$\partial p/\partial t$	ΔU	ΔV
1/0/0	7.6e-05	2.0e-07	4.4e-07	3.8e-07	2.9e-06	8.6e-07	6.2e-07
2/2/3	6.2e-13	9.8e-16	5.7e-15	5.0e-15	5.1e-14	1.5e-14	2.3e-14
2/3/3	5.5e-13	1.0e-15	5.9e-15	5.3e-15	2.5e-14	1.5e-14	2.3e-14
3/1/1	5.2e-13	8.5e-16	5.0e-15	4.6e-15	4.0e-14	1.4e-14	2.0e-14
3/2/1	6.0e-13	7.5e-16	5.3e-15	4.8e-15	3.7e-14	1.4e-14	2.1e-14
3/2/2	8.1e-13	1.0e-15	4.6e-15	4.6e-15	3.5e-14	1.3e-14	1.9e-14
3/2/3	5.8e-13	2.5e-13	6.0e-15	5.3e-15	4.9e-14	1.6e-14	2.3e-14
3/3/2	8.7e-13	1.3e-15	4.9e-15	4.6e-15	4.4e-14	1.4e-14	2.1e-14
3/3/3	1.1e-12	2.6e-15	6.0e-15	5.4e-15	6.5e-14	1.6e-14	2.4e-14
3/1/2	5.4e-13	1.4e-15	4.9e-15	4.5e-15	7.4e-14	1.3e-14	2.0e-14
3/1/3	1.6e-12	1.8e-15	6.4e-15	5.6e-15	5.4e-14	1.6e-14	2.4e-14
2/1/3	6.9e-13	1.3e-15	5.6e-15	5.0e-15	4.2e-14	1.4e-14	2.3e-14

solutions were also quite accurate. The computed differences between the analytical and numerical fluxes were quite small; the divergence and pressure time derivatives had similar trends. The residuals from the continuity equation were indeed the last (compared to the momentum equations residuals) to reach convergence tolerance. Once the continuity violation had been reduced below 0.0001, no discernible difference in the flux field from its analytical solution was observed. In addition, once this level of convergence was reached, the gradient of the computed pressure field was observed to be approximately zero as well.

Several hundred iterations were required to drive continuity violations to below the assigned convergence tolerance for the fine grid-only simulation. This trend was observed as well for the mass conservation simulations documented in Chapter 7.

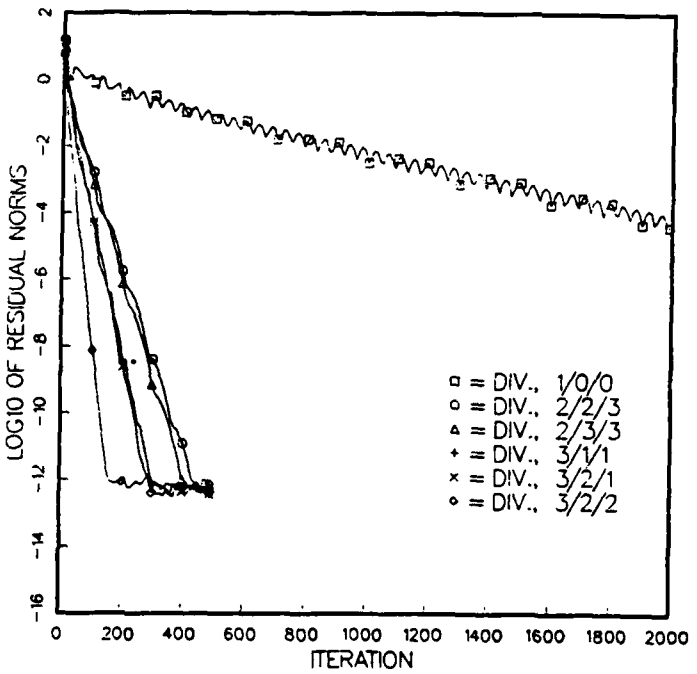


Figure 8.3a. Fine-grid-only and 5 of 11 multigrid simulations

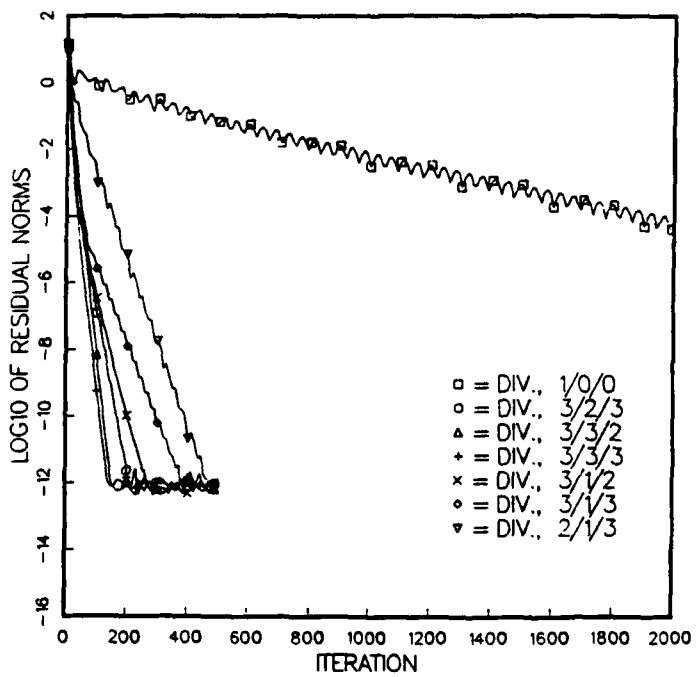


Figure 8.3b. Fine-grid-only and remaining multigrid simulations

Figure 8.3. Convergence history for divergence of velocity, 21-by-21 grid, Couette flow, Re = 100, multiple grid runs

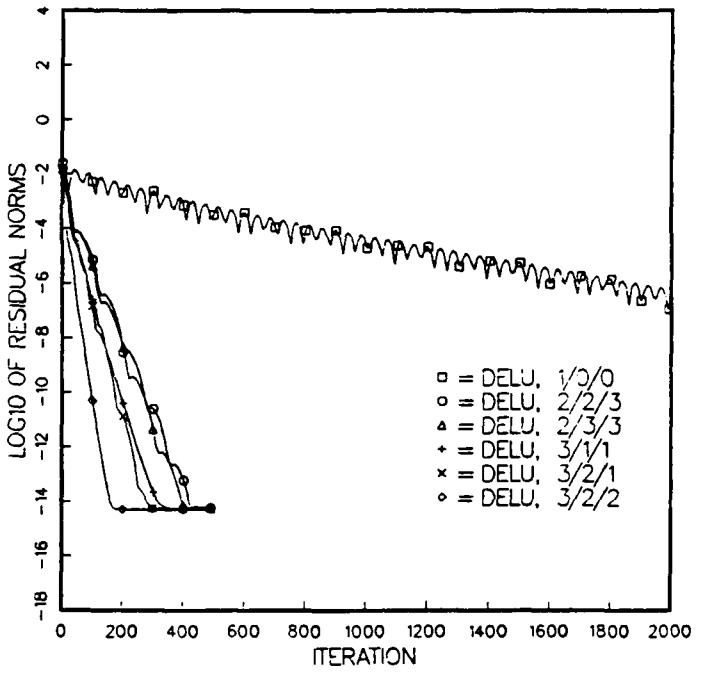


Figure 8.4a. Fine-grid-only and 5 of 11 multigrid simulations

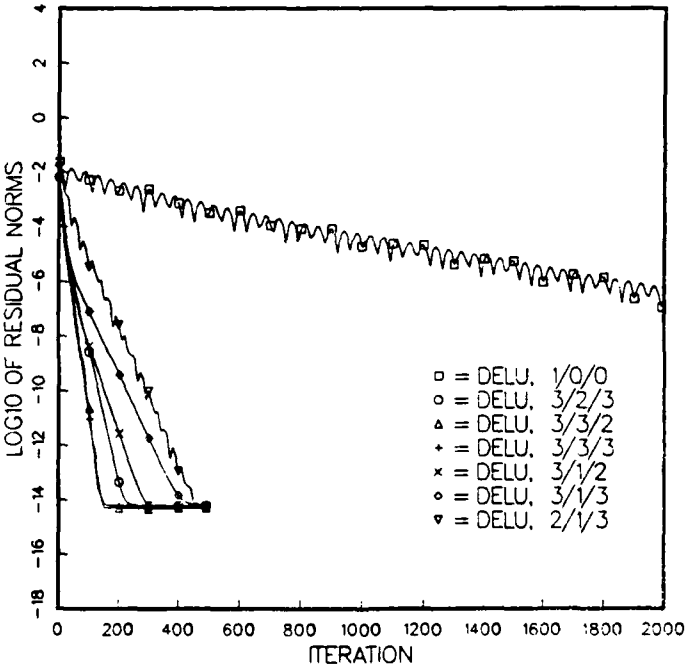


Figure 8.4b. Fine-grid-only and remaining multigrid simulations

Figure 8.4. Convergence history for U flux, 21-by-21 grid, Couette flow, Re = 100, multiple grid runs

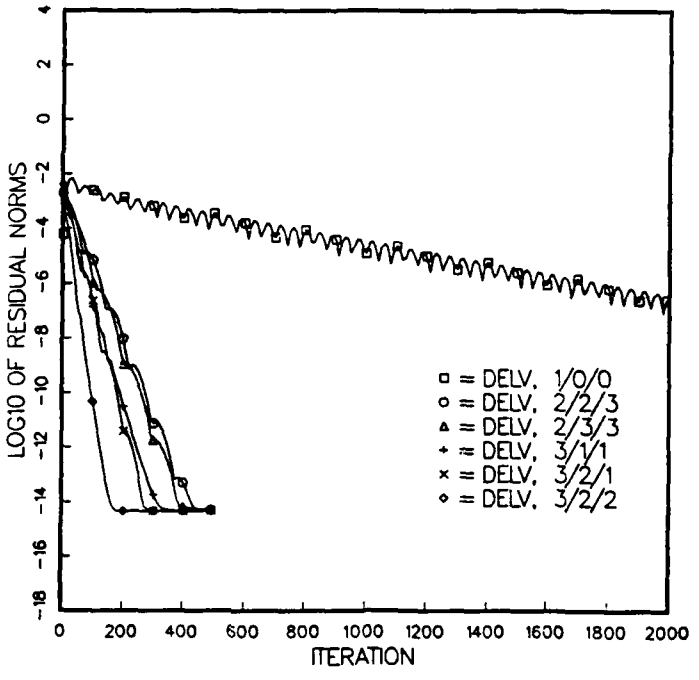


Figure 8.5a. Fine-grid-only and 5 of 11 multigrid simulations

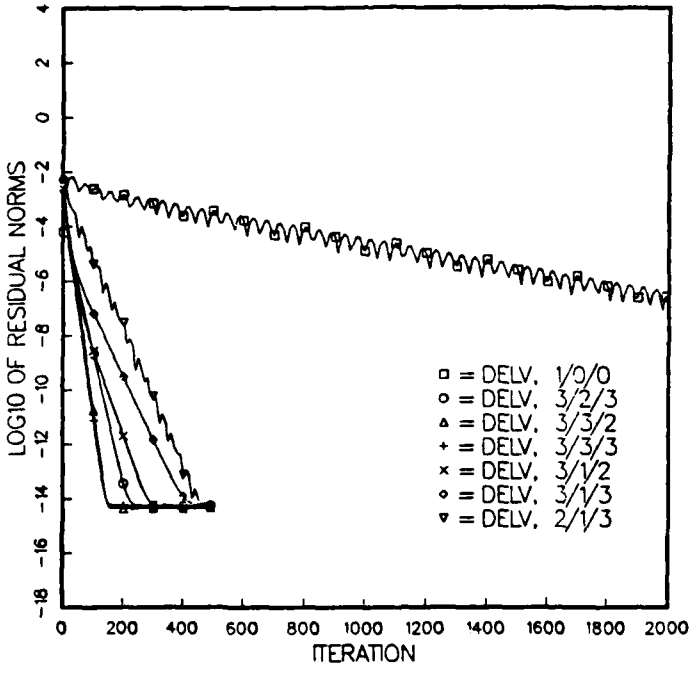


Figure 8.5b. Fine-grid-only and remaining multigrid simulations

Figure 8.5. Convergence history for V flux, 21-by-21 grid, Couette flow, Re = 100, multiple grid runs

The fine-grid solution (designated as 1/0/0) was observed to fluctuate as it proceeded to the steady state. These fluctuations are typical of the MacCormack predictor-corrector scheme for reasons stated in the previous chapter. However, use of the multigrid approach clearly alleviated much of this problem. The convergence histories for the multigrid runs were much smoother than the fine grid-only history.

Given in Table 8.2 is a comparison of the relative run times to convergence for this flow scenario with no multigrid (1 fine grid only) and with the multigrid algorithm added. The speed-up factors given represent the ratio of the relative run time for the fine grid-only run divided by that of the given multigrid scenarios. The addition of the multigrid scheme significantly quickened the convergence of the model test case in most cases. In three cases, however, the multigrid scheme was less than three times as efficient in reaching the assigned convergence tolerance as the fine grid alone. In all three cases, only one additional level of coarse grid was added with the result that the overhead of setting up multigrid operations was a relatively large percentage of the reduction in iteration time saved as compared with the finest grid-only calculation. Thus, care must be exercised in the implementation and utilization of multigrid.

Table 8.2 also shows the use of multiple rather than single relaxation sweeps on each grid enhanced convergence on the 21-by-21 grid for the three-grid test cases. In fact, the same general scenarios showed maximized speedup as did for the mass conservation scenarios reported in Chapter 7. The optimum number of fine and coarse relaxation sweeps is still not clear from these results. However, it is obvious

Table 8.2
Comparison of Convergence Properties for
Multiple Grid Runs: Couette Flow, 21-by-21 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 100

# Grids	# Fine Sweeps	# Coarse Sweeps	# Iterations	Relative Time to Con. Tolerance	Speed Up Factor
1	N/A	N/A	1770	45.5	1.00
2	2	3	130	16.7	2.73
2	3	3	130	19.8	2.30
3	1	1	90	8.9	5.10
3	2	1	90	11.1	4.10
3	2	2	40	5.6	8.16
3	2	3	50	7.8	5.84
3	3	2	50	8.2	5.56
3	3	3	35	6.3	7.22
3	1	2	40	4.6	9.87
3	1	3	50	6.6	6.91
2	1	3	150	15.6	2.91

of that the use some level of multiplicity for either fine or coarse relaxation is in order.

The results presented above again illustrate well the need use as many coarse grids as possible given the resolution of the finest grid. The maximum number of grids which could be employed for this scenario was 3.

8.2.2 Results for the 21-by-21 grid, Re = 400

Results of the 21-by-21 grid simulations are given in Tables 8.3 and 8.4; convergence histories for these scenarios are given in Figures 8.6-8.8. For this test case, convergence was again defined as having been reached when the maximum divergence was less than 0.0001. The results indicate that the use of the multigrid approach is warranted for this problem based on enhanced use of computer resources. Table 8.3 illustrates clearly the accuracy of the numerical scheme. The final level of convergence for this case is somewhat poorer than that for the

lesser Reynolds number case above; however, the convergence is still outstanding. The results again bear out the need to employ as many coarse grids as possible as shown by the enhanced efficiency of the 3-grid runs over their 2-grid counterparts. And, though not shown, the pressure gradient was observed to be less than 10^{-5} for all runs which had divergence norms of less than 10^{-4}.

A number of interesting aspects of the simulation of this test case are illustrated in Table 8.3. The simulations for the Re = 400 case converged somewhat more slowly than the Re = 100 simulations. This is also mirrored in Table 8.4 which shows that the speedups for the higher Reynolds number case were generally lower than the same Re = 100 scenario (as listed in Table 8.2). Most of the general trends listed previously are exhibited for the higher Reynolds number case. Unlike the lower Reynolds number case presented in the previous section, some

Table 8.3
Summary of Convergence Norms for
Multiple Grid Runs: Couette Flow, 21-by-21 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 400

		Average			Maximum		
Run	$\nabla \cdot u$	$\partial p/\partial t$	ΔU	ΔV	$\partial p/\partial t$	ΔU	ΔV
1/0/0	1.6e-06	2.2e-09	1.0e-08	4.7e-09	4.7e-09	2.5e-08	1.1e-08
2/2/3	2.3e-12	2.4e-14	1.8e-14	1.6e-14	5.9e-13	5.1e-14	5.2e-14
2/3/3	1.7e-12	1.2e-14	6.3e-15	5.5e-15	2.7e-13	2.4e-14	1.9e-14
3/1/1	4.8e-08	6.8e-11	1.1e-09	9.3e-10	2.1e-09	2.6e-09	3.7e-09
3/2/1	6.8e-13	4.7e-16	5.1e-15	4.7e-15	2.7e-14	1.5e-14	2.1e-14
3/2/2	1.2e-12	2.4e-15	2.5e-14	2.2e-14	6.3e-14	5.9e-14	8.3e-14
3/2/3	4.0e-11	1.6e-13	8.4e-13	4.2e-12	9.6e-11	2.2e-11	2.7e-11
3/3/2	1.0e-12	6.3e-16	4.6e-15	4.3e-15	2.2e-14	1.4e-14	2.0e-14
3/3/3	1.5e-12	2.0e-15	9.6e-15	8.8e-15	4.6e-14	2.4e-14	3.9e-14
3/1/2	2.9e-08	4.4e-11	6.3e-10	5.1e-10	1.3e-09	1.5e-09	1.9e-09
3/1/3	1.9e-08	1.5e-11	1.9e-10	1.1e-10	5.4e-10	6.1e-10	6.4e-10
2/1/3	5.0e-08	1.2e-10	2.6e-09	2.2e-09	3.2e-09	5.9e-09	8.6e-09

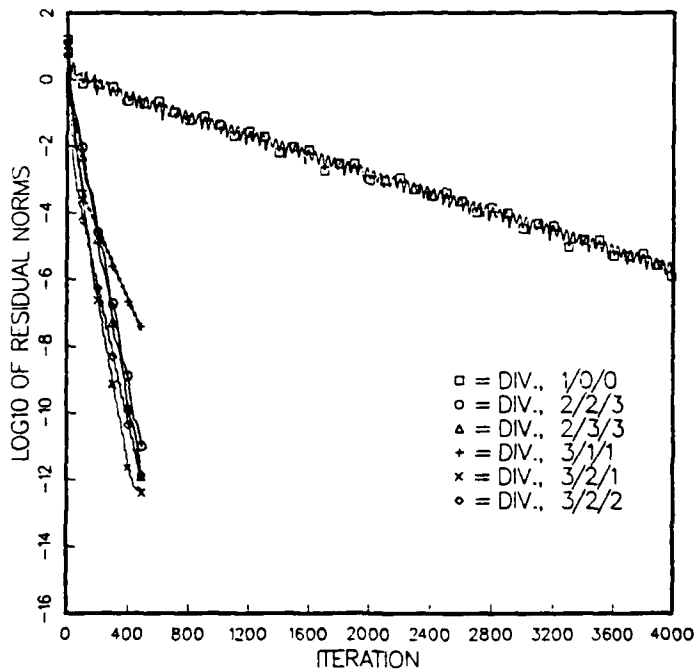


Figure 8.6a. Fine-grid-only and 5 of 11 multigrid simulations

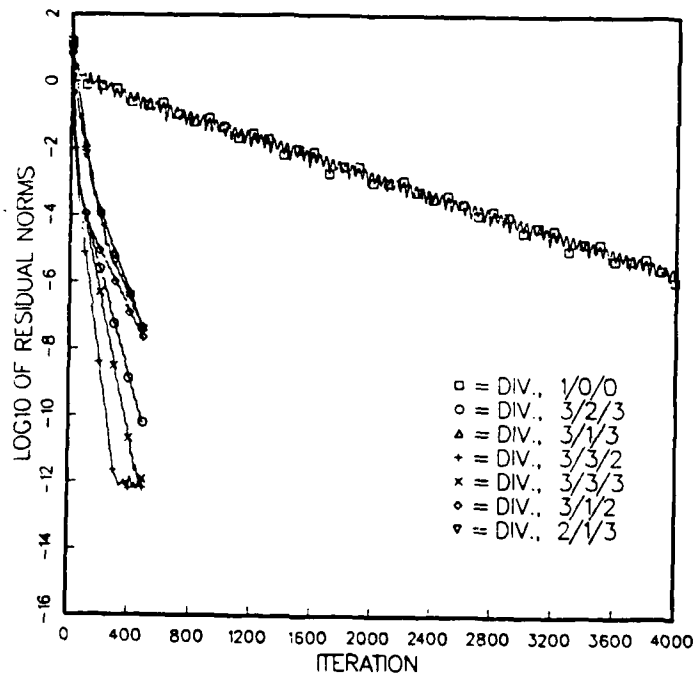


Figure 8.6b. Fine-grid-only and remaining multigrid simulations

Figure 8.6. Convergence history for divergence of velocity, 21-by-21 grid, Couette flow, Re = 400, multiple grid runs

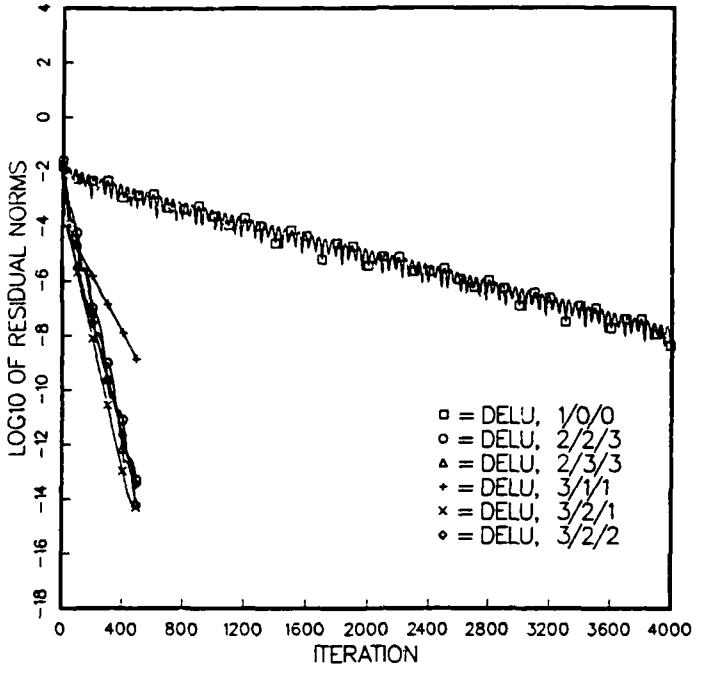


Figure 8.7a. Fine-grid-only and 5 of 11 multigrid simulations

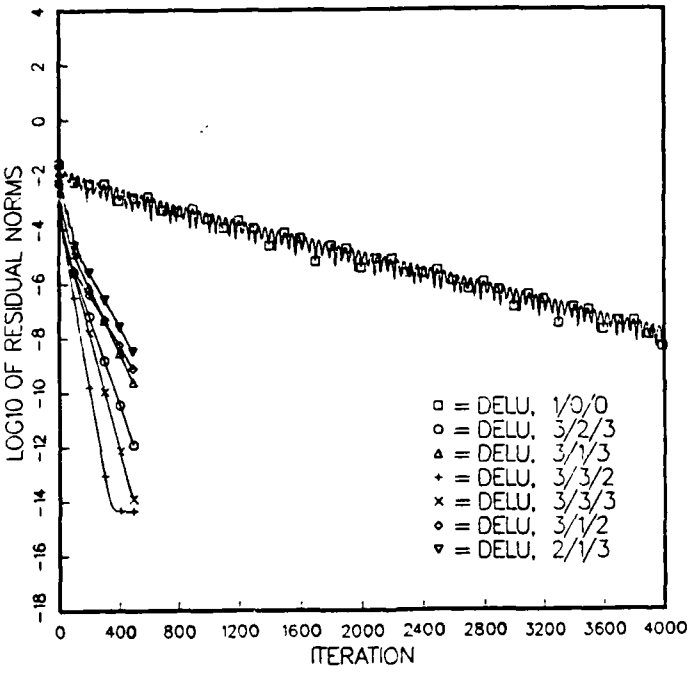


Figure 8.7b. Fine-grid-only and remaining multigrid simulations

Figure 8.7. Convergence history for U flux, 21-by-21 grid, Couette flow, Re = 400, multiple grid runs

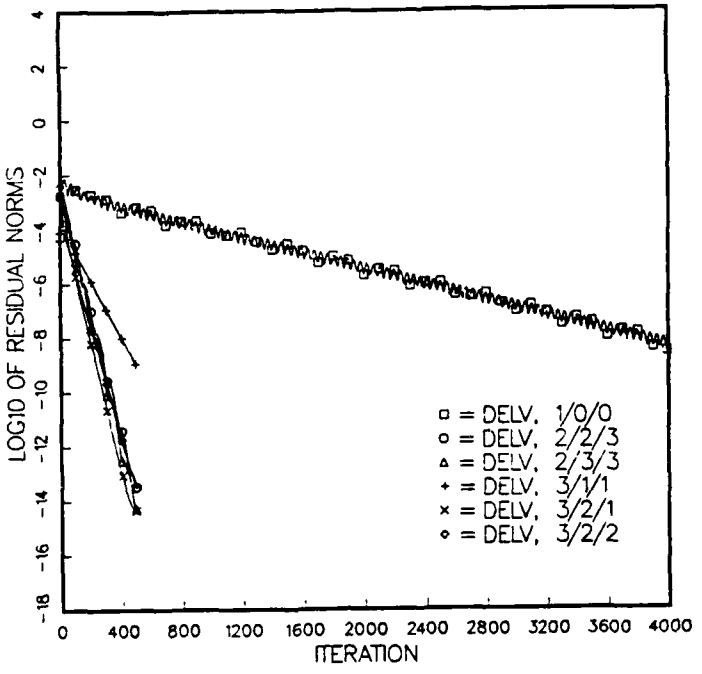


Figure 8.8a. Fine-grid-only and 5 of 11 multigrid simulations

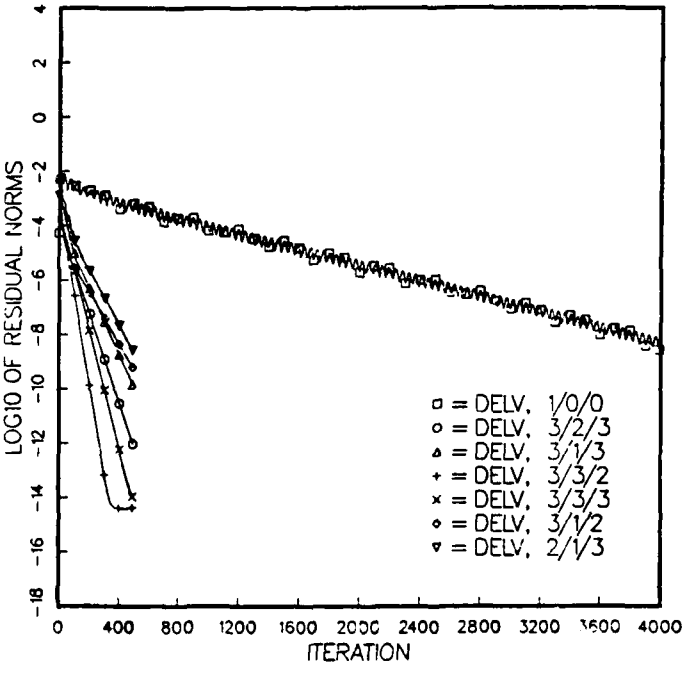


Figure 8.8b. Fine-grid-only and remaining multigrid simulations

Figure 8.8. Convergence history for V flux, 21-by-21 grid, Couette flow, Re = 400, multiple grid runs

Table 8.4
Comparison of Convergence Properties for
Multiple Grid Runs: Couette Flow, 21-by-21 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 400

# Grids	# Fine Sweeps	# Coarse Sweeps	# Iterations	Relative Time to Con. Tolerance	Speed Up Factor
1	N/A	N/A	2700	69.5	1.00
2	2	3	180	23.1	3.01
2	3	3	160	24.7	2.82
3	1	1	140	13.9	5.01
3	2	1	90	11.1	6.27
3	2	2	80	11.2	6.23
3	2	3	80	12.5	5.58
3	3	2	60	9.8	7.08
3	3	3	90	16.2	4.29
3	1	2	100	11.5	6.03
3	1	3	200	26.3	2.64
2	1	3	190	19.9	3.49

of the 3-grid, one relaxation sweep on the finest grid cases exhibited poorer final convergence at 500 iterations than did the other scenarios simulated. The rate at which these 3-grid scenarios reached a divergence norm of 0.0001 was still, except for the 3/1/3 case, relatively good. These 3-grid simulations were not, contrary to previous results, the best of the best in terms of speedup.

The reasons for these 3-grid results are not completely clear. In the case of the 3/1/3 run, the combination of higher Reynolds number, 1 relaxation sweep on the finest grid, 3 coarse-grid relaxation sweeps, and the initial conditions used may have resulted in the generation of subtle errors for this case that defeated some of the error smoothing abilities of the multigrid scheme. As shown in Figure 8.6, the convergence of the divergence for the 3/1/3 scenario was not arrested; rather, it was continuing to converge at 500 cycles. The rate of

convergence for this scenario, however, was much lower than several of the other runs. This problem may be related, in part, to the start-up problems discussed for the nonorthogonal case study in Chapter 7. Recall that the 3/1/3 nonorthogonal mass conservation scenario required an excessive amount of under-relaxation to achieve stable results. Although not as severe, the orthogonal 3/1/3 case in Chapter 7 also required some under-relaxation. For the nonorthogonal case, the speedup was negligible; it was relatively good for the orthogonal case, though. These results, and the fact that the Re = 100 Couette flow results of the previous section required a small amount of under-relaxation for the 3/1/3 case, suggest that the numerical scheme is not as computational efficient or stable for this grid/sweep scenario as others. The results also lead to the hypothesis that the combination of maximum number of grids/one finest-grid relaxation sweep/maximum number of coarse-grid relaxation sweeps may be an inappropriate scenario for this scheme's formulation and the selected initial conditions. This hypothesis will be investigated further in subsequent sections of this chapter.

An interesting aspect of the results presented in Chapters 7 and 8 to this point is that, while the specific ordering of the better scenarios in terms of speedup is different from case to case, the same general grid/relaxation sweep scenarios have been optimal thus far. The 3/1/2, 3/2/3, 3/1/1, 3/1/3 (to some extent), 3/2/1, 3/2/2, 3/3/3, and 3/3/2 scenarios represent the extent of the 4 top speedup scenarios for the case studies discussed to date. This information is hardly definitive however.

8.2.3 Results for the 41-by-41 grid, Re = 100

To shed further light on the trends given above, and to examine the effects of increasing the number of coarse grids employed, the finest grid's resolution was effectively doubled. A maximum of 4 grids was then utilized. Results for these simulations are given in Tables 8.5 and 8.6 as well as Figures 8.9-8.11.

The results for this scenario are accurate. As shown in Table 8.5, all of the runs converged very well with the possible exception of the 2/2/3 simulation. The need to use as many grids as possible is even more vividly brought out by these results than those previous as illustrated by the poor showing of the 2-grid runs. Additionally, the 4-grid runs were generally superior in terms of speedup than the 3-grid simulations.

The level of speedup for this case is considerably higher than for the previous 21-by-21 grid cases. This result points out the true utility of multigrid lies in the convergence acceleration of large problems being solved over a solution space discretized with many nodes. This is very encouraging when considered in the future context of three-dimensional solutions.

The need to highly under-relax for certain scenarios was observed for this model case. The 3/2/3, 3/1/3, 4/1/2, 4/1/3, 4/2/2, 4/2/3, 4/3/2, and 4/3/3 scenarios for the 41-by-41 grid runs required varying amounts of under-relaxation for computational stability and proper convergence acceleration. The relaxation coefficient, which multiplies the time-step calculation and was usually 0.9 for most simulations, had values ranging from 0.55 to 0.20 for these runs. The reason for this undoubtedly lies in the connection between the initial conditions

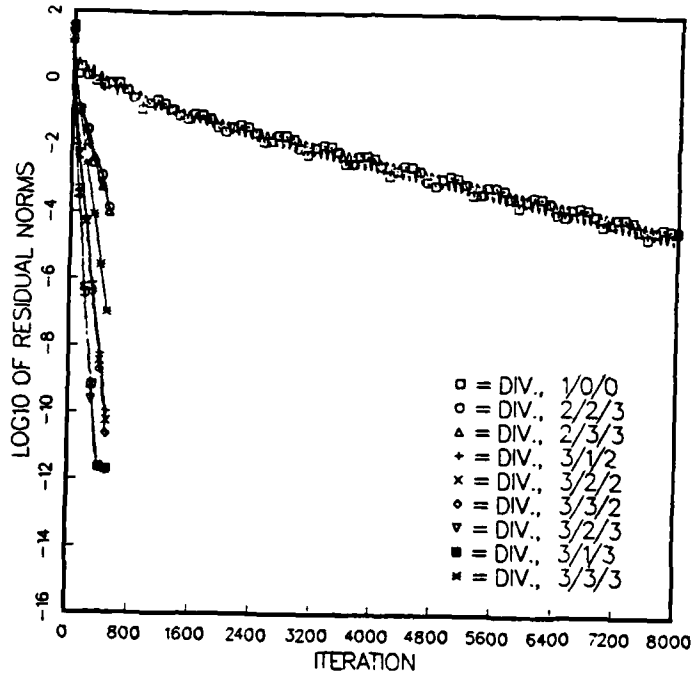


Figure 8.9a. Fine-grid-only and 8 of 16 multigrid simulations

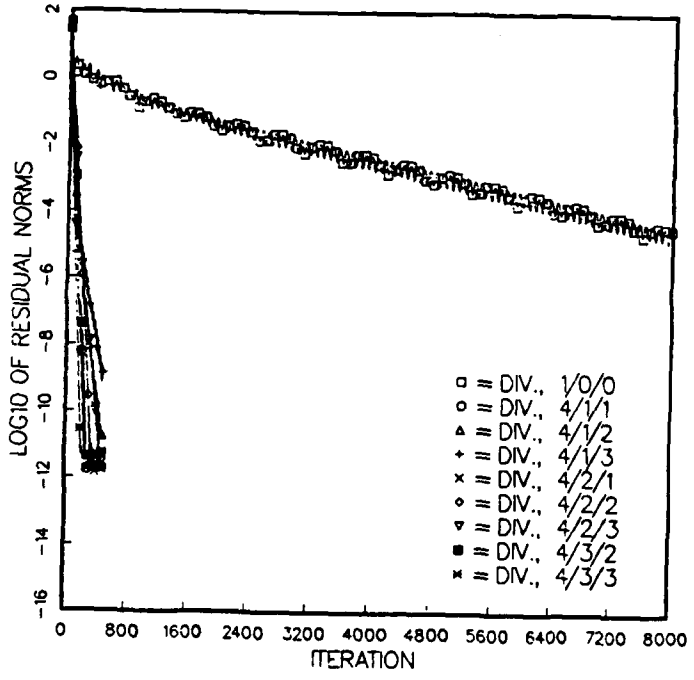


Figure 8.9b. Fine-grid-only and remaining multigrid simulations

Figure 8.9. Convergence history for divergence of velocity, 41-by-41 grid, Couette flow, Re = 100, multiple grid runs (Continued)

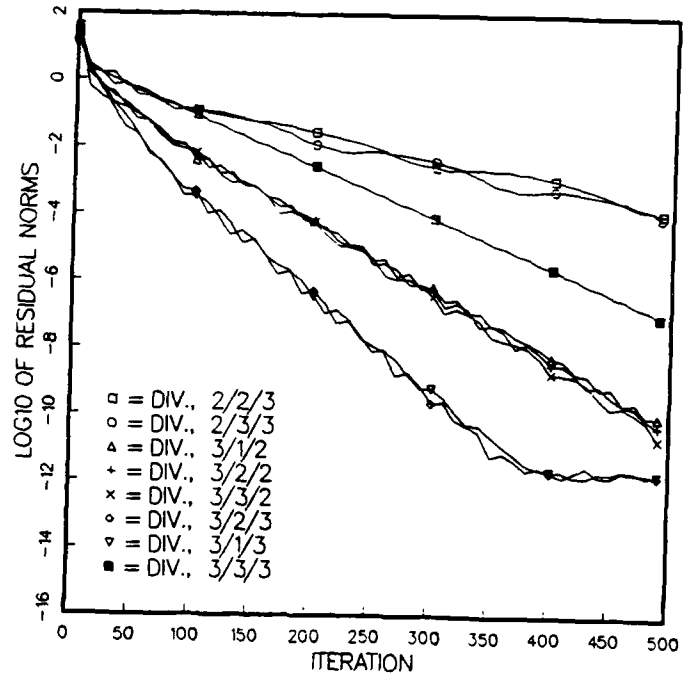


Figure 8.9c. Inset of 8.9a showing multigrid simulations

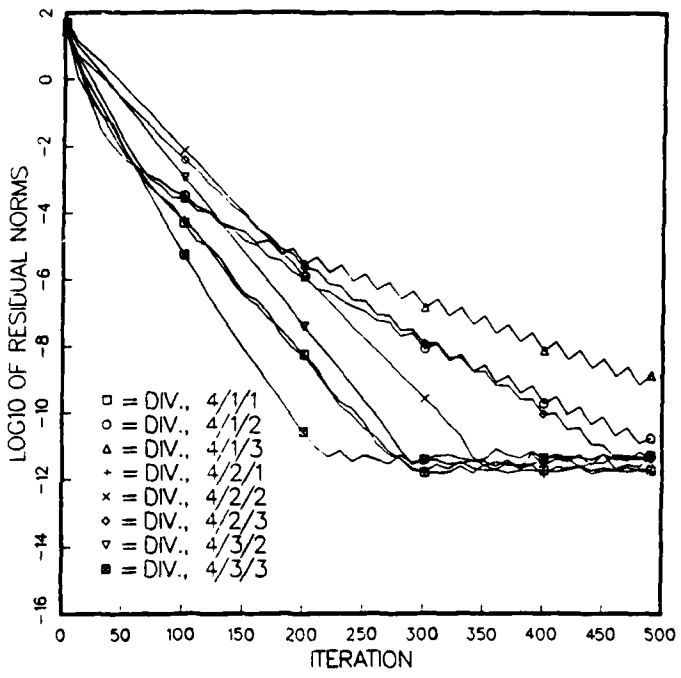


Figure 8.9d. Inset of 8.9b showing multigrid simulations

Figure 8.9 (Concluded)

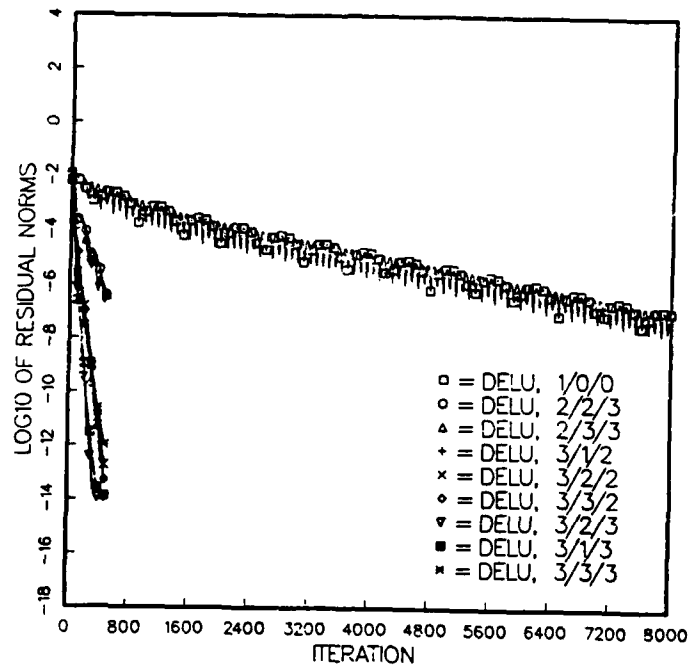


Figure 8.10a. Fine-grid-only and 8 of 16 multigrid simulations

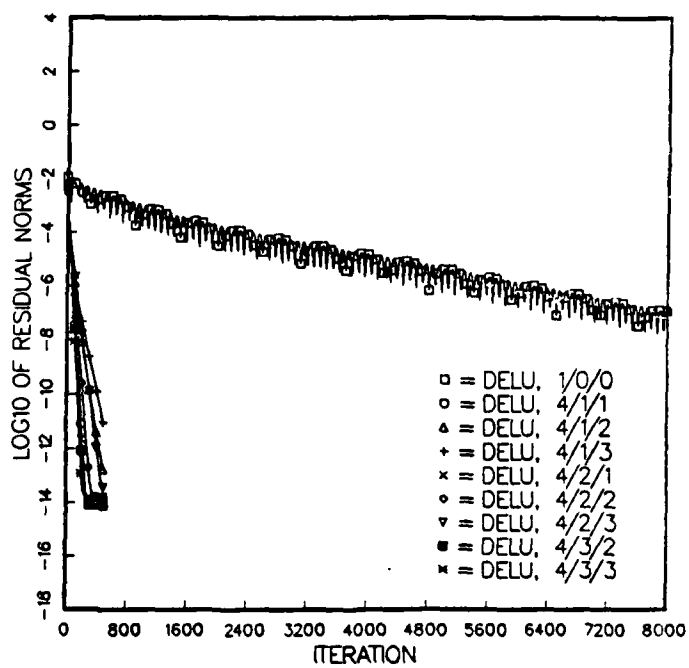


Figure 8.10b. Fine-grid-only and remaining multigrid simulations

Figure 8.10. Convergence history for U flux, 41-by-41 grid, Couette flow, Re = 100, multiple grid runs (Continued)

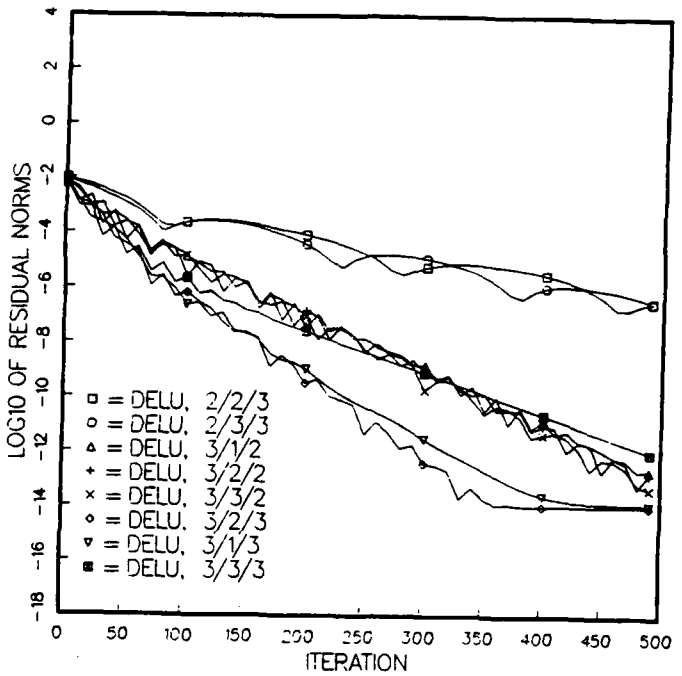


Figure 8.10c. Inset of 8.10a showing multigrid simulations

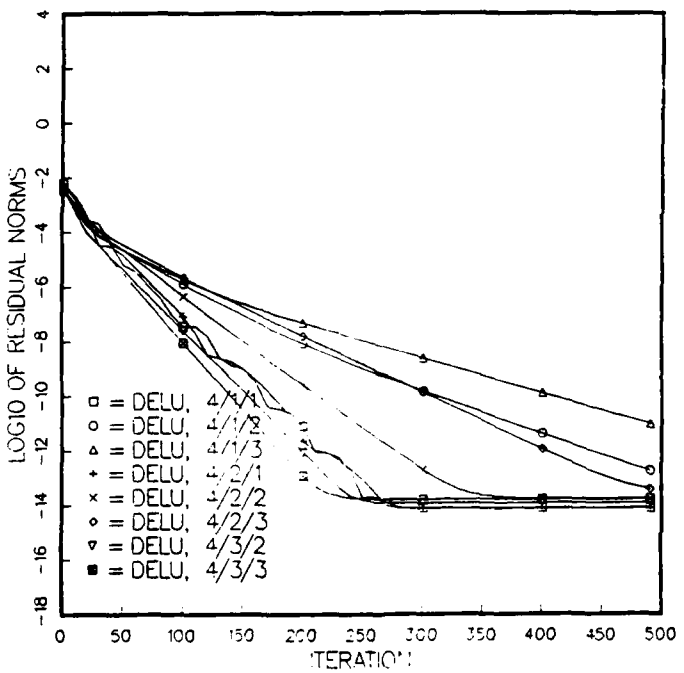


Figure 8.10d. Inset of 8.10b showing multigrid simulations

Figure 8.10 (Concluded)

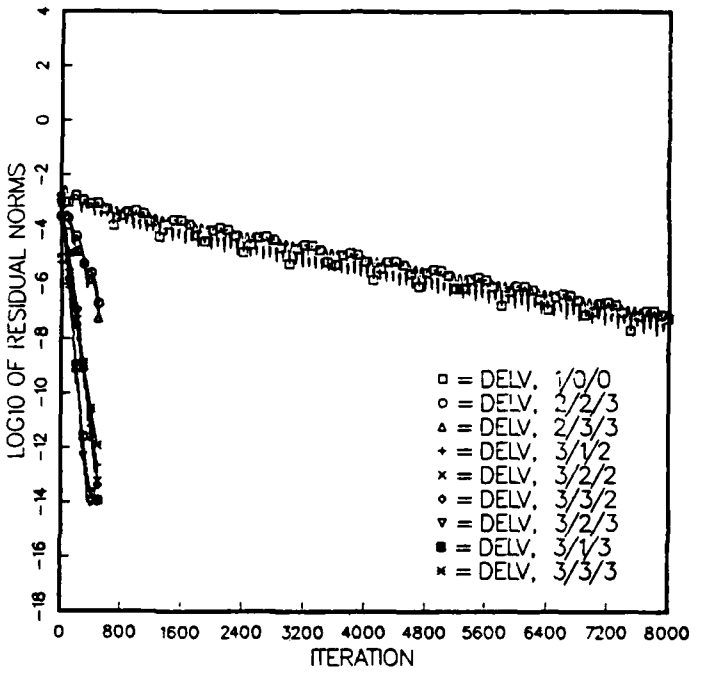


Figure 8.11a. Fine-grid-only and 8 of 16 multigrid simulations

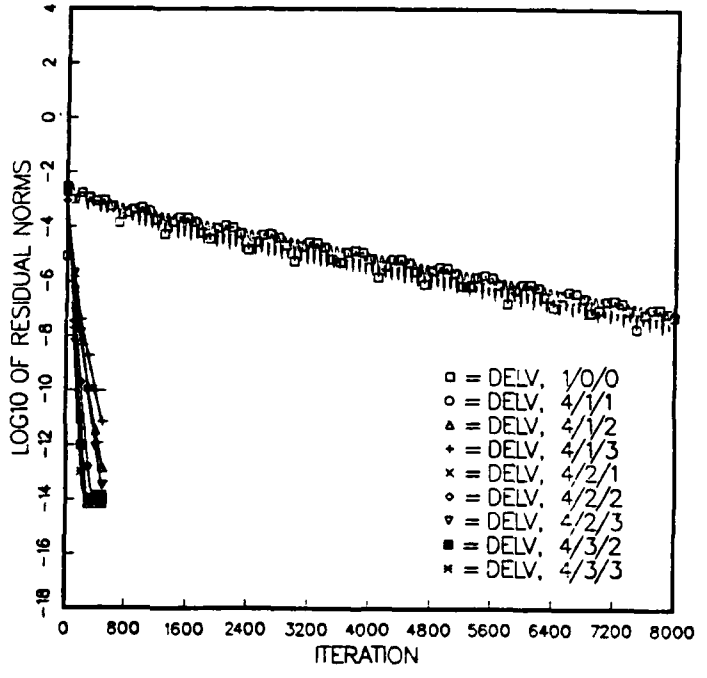


Figure 8.11b. Fine-grid-only and remaining multigrid simulations

Figure 8.11. Convergence history for V flux, 41-by-41 grid, Couette flow, Re = 100, multiple grid runs (Continued)

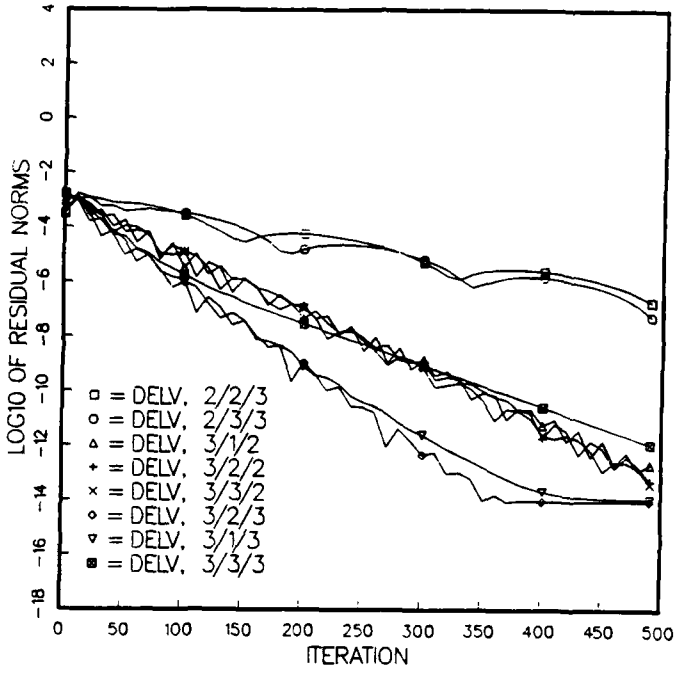


Figure 8.11c. Inset of 8.11a showing multigrid simulations

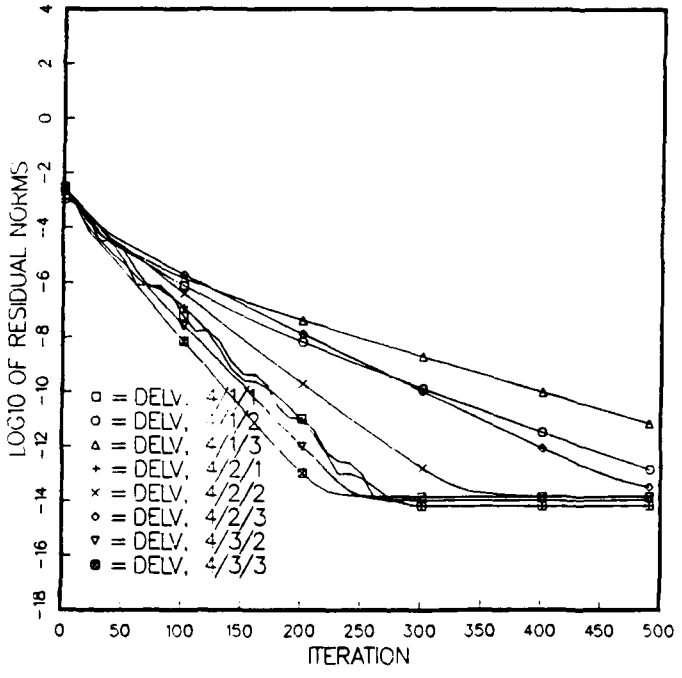


Figure 8.11d. Inset of 8.11b showing multigrid simulations

Figure 8.11 (Concluded)

Table 8.5
Summary of Convergence Norms for
Multiple Grid Runs: Couette Flow, 41-by-41 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 100

		Average			Maximum		
Run	$\nabla \cdot u$	$\partial p/\partial t$	ΔU	ΔV	$\partial p/\partial t$	ΔU	ΔV
1/0/0	3.1e-05	3.5e-08	1.0e-07	3.3e-08	7.7e-07	1.9e-07	7.2e-08
2/2/3	1.7e-04	1.4e-06	5.7e-07	1.2e-07	3.0e-05	1.1e-06	3.5e-07
2/3/3	1.0e-04	9.2e-07	3.3e-07	7.6e-08	2.0e-05	6.1e-07	2.1e-07
3/1/2	4.3e-11	4.4e-13	1.3e-13	2.8e-14	9.3e-12	2.8e-13	9.1e-13
3/2/2	5.3e-11	3.3e-13	5.5e-14	1.0e-13	6.4e-12	1.4e-13	1.8e-13
3/2/3	3.2e-12	2.4e-15	1.1e-14	9.7e-15	1.3e-13	2.9e-14	5.1e-14
3/3/2	3.0e-11	3.4e-13	8.2e-14	4.9e-14	7.3e-12	1.6e-13	1.0e-13
3/3/3	8.5e-08	4.2e-12	7.5e-13	9.4e-13	8.9e-10	3.1e-11	4.9e-11
3/1/3	1.9e-12	3.5e-15	1.3e-14	1.1e-14	1.3e-13	3.4e-14	5.9e-14
4/1/1	1.9e-12	1.0e-15	7.8e-15	6.5e-15	7.2e-14	1.9e-14	3.5e-14
4/1/2	1.5e-11	8.3e-14	1.3e-13	1.0e-13	1.1e-13	4.0e-13	5.0e-13
4/1/3	4.6e-10	6.0e-12	6.7e-12	5.1e-12	8.6e-11	2.2e-11	2.4e-11
4/2/1	4.2e-12	8.6e-16	7.5e-15	6.4e-15	6.3e-14	1.9e-14	3.5e-14
4/2/2	2.8e-12	1.2e-14	1.6e-14	1.4e-14	3.1e-13	4.3e-14	7.4e-14
4/2/3	7.7e-12	2.8e-14	3.3e-14	2.8e-14	5.2e-13	8.6e-14	1.5e-13
4/3/2	4.5e-12	7.2e-15	1.2e-14	1.1e-14	2.4e-13	3.2e-14	5.4e-14
4/3/3	4.7e-12	1.3e-14	1.6e-14	1.4e-14	2.4e-13	4.2e-14	7.3e-14

chosen, the problem being solved, and the number of grids/number of relaxation sweeps selected. This brings into question whether the initial conditions being used thus far in this report, which set the initial flux and pressure values in the field to zero, are worth utilizing.

It is probably true that somewhat different results from those shown could be obtained for differing initial conditions. The initial conditions used herein, however, were chosen because they are simple to utilize and require no pre-conditioning or pre-simulation. There is something quite attractive about having algorithm which, for the simple initial conditions employed, will quickly achieve the correct solution.

Table 8.6
Comparison of Convergence Properties for
Multiple Grid Runs: Couette Flow, 41-by-41 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 100

# Grids	# Fine Sweeps	# Coarse Sweeps	# Iterations	Relative Time to Con. Tolerance	Speed Up Factor
1	N/A	N/A	6400	583.2	1.00
2	2	3	500	234.0	2.49
2	3	3	500	227.0	2.11
3	1	2	175	73.5	7.93
3	2	2	175	80.9	6.56
3	2	3	120	68.2	8.55
3	3	2	175	104.0	5.61
3	3	3	280	183.1	3.19
3	1	3	120	57.8	10.09
4	1	1	95	36.5	15.99
4	1	2	120	54.5	10.70
4	1	3	120	62.9	9.27
4	2	1	95	44.7	13.06
4	2	2	155	83.7	6.97
4	2	3	155	94.6	7.13
4	3	2	125	78.3	7.45
4	3	3	80	55.7	10.47

The need to under-relax, still, is a difficult one to broach. Given that the zero-start is being used is for simplicity and efficiency, the need to evaluate numerous under-relaxation coefficients until enhanced performance and stability are reached undermines this reasoning. It is interesting that the top two scenarios in terms of speedup for each of the cases reported herein generally required no additional under-relaxation; the coefficient value of 0.9 was sufficient to achieve quite good results. Further, these optimum scenarios usually were combinations of the maximum number of grids allowable, either one or two finest-grid relaxation sweeps, and either one or two coarse-grid relaxation sweeps. The validity of this assessment will be evaluated in the next case study.

The combination of maximum number of grids allowable, one finest-grid relaxation sweep, and maximum number of coarse-grid relaxation sweeps appears somewhat less efficient computationally than the other grid/sweeps combinations. This supports the results of simulation this scenario for other case studies. The 4/1/3 run had a median speedup observed among those for the 4-grid simulations as shown in Table 8.6. This suggests that this combination is not one of the more efficient for the variety of cases studied herein.

8.3 RESULTS OF STATIONARY VISCOUS CHANNEL FLOW

In order to gain more insight into the proposed trends given above, and to evaluate the efficacy of the numerical scheme for a differing model test case, Couette flow with a pressure gradient was simulated. The conditions simulated are discussed at the beginning of this chapter. This model test case, with its two no-slip horizontal boundaries, a slip inlet, and a slip outlet, behaves very much like riverine flow within a straight reach.

8.3.1 Results of the 21-by-21 grid, Re = 100

Results of these simulations are given in Tables 8.7 and 8.8, and in Figures 8.12-8.14. The conditions simulated were chosen based on the trends listed above. Only the better 3-grid multigrid scenarios were run along with the fine grid-only condition. Table 8.7 shows that, the 1/0/0 and multigrid runs produced excellent results. The divergence was consistently very small for all runs, as were the deviations of the predicted fluxes from their analytical counterparts. In addition, the time derivative of pressure was very small.

Table 8.7
Summary of Convergence Norms for
Multiple Grid Runs: Stationary, Viscous Flow,
21-by-21 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 100

		Average			Maximum		
Run	$\nabla \cdot u$	$\partial p/\partial t$	ΔU	ΔV	$\partial p/\partial t$	ΔU	ΔV
1/0/0	4.5e-06	7.6e-09	2.3e-09	2.9e-08	1.2e-07	5.2e-08	1.5e-08
3/1/1	6.6e-13	8.7e-16	5.1e-15	2.3e-15	2.8e-14	1.4e-14	1.4e-14
3/1/2	6.8e-13	1.1e-15	5.0e-15	2.2e-15	2.9e-14	1.2e-14	1.3e-14
3/2/2	6.1e-13	1.4e-15	5.2e-15	2.3e-15	4.7e-14	1.3e-14	1.4e-14
3/2/1	4.8e-13	9.7e-16	5.4e-15	2.3e-15	2.6e-14	1.4e-14	1.5e-14
3/2/3	1.5e-12	2.4e-15	5.0e-15	2.2e-15	9.7e-14	1.2e-14	1.3e-14
3/3/2	2.6e-12	5.8e-15	5.2e-15	2.1e-15	1.4e-13	1.3e-14	1.4e-14
3/3/3	1.0e-12	2.1e-15	5.1e-15	2.1e-15	5.7e-14	1.3e-14	1.3e-14
3/1/3	8.5e-13	2.0e-15	4.7e-15	2.3e-15	5.4e-14	1.1e-14	1.2e-14

Given in Table 8.8 are the speedups for these simulations. The same general simulations provided the better speedups just as in previous 21-by-21 grid runs. The 3/3/2 run provided little convergence acceleration for the computer resources utilized. This simulation, and the 3/2/3 run, required some additional relaxation beyond that needed by the other runs. Both used an under-relaxation coefficient of 0.8; a value of 0.9 was used for the remaining runs. While these coefficients are very similar, their difference does point toward the possibility that the 3/3/2 run actually needed even more under-relaxation for efficient convergence. No effort was made to optimize this under-relaxation coefficient for the 3/3/2 or 3/2/3 runs.

Evaluating the analytical expressions in Equations 8.1-8.4 for the conditions of this case, it was found that the $\partial p/\partial x$ term (recalling that p is actually pressure divided by density) for this problem should be 0.0040. This value was reached for all simulations having continuity violations of less than 0.0001. All of the simulations produced the

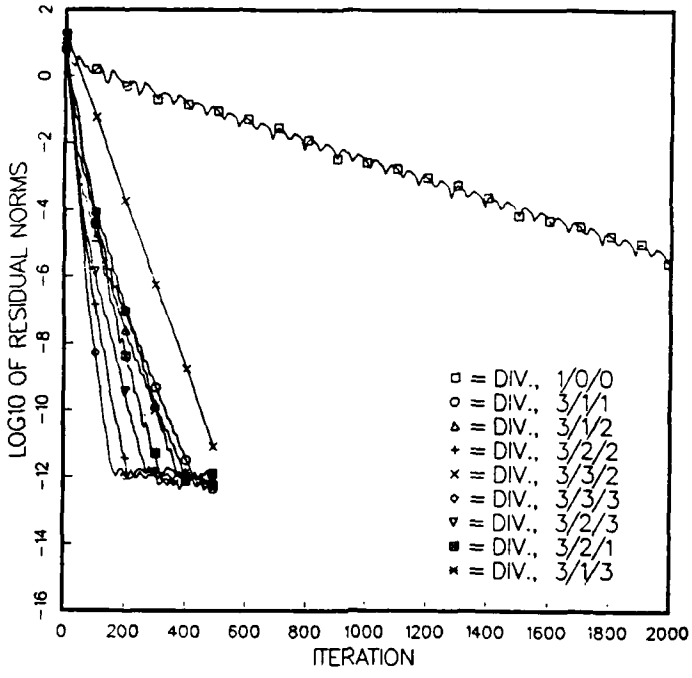


Figure 8.12. Convergence history for divergence of velocity, 21-by-21 grid, stationary, viscous flow, Re = 100, multiple grid runs

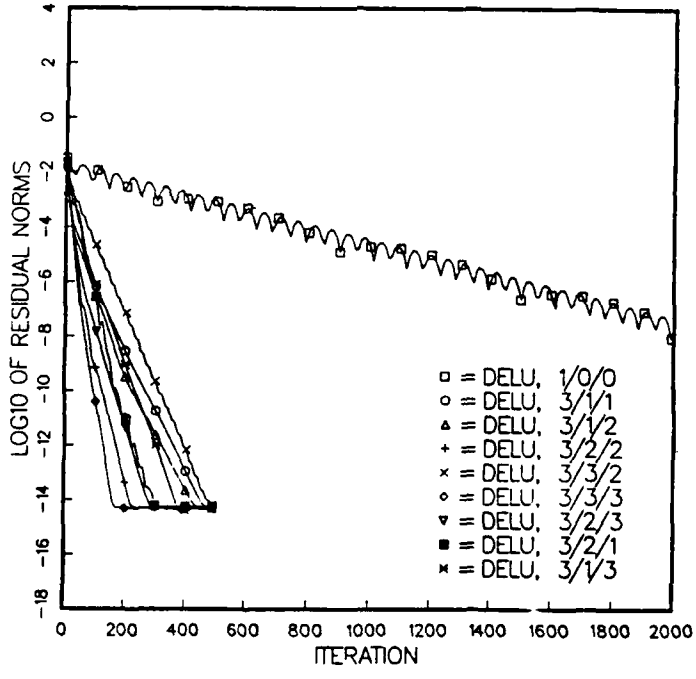


Figure 8.13. Convergence history for U flux, 21-by-21 grid, stationary, viscous flow, Re = 100, multiple grid runs

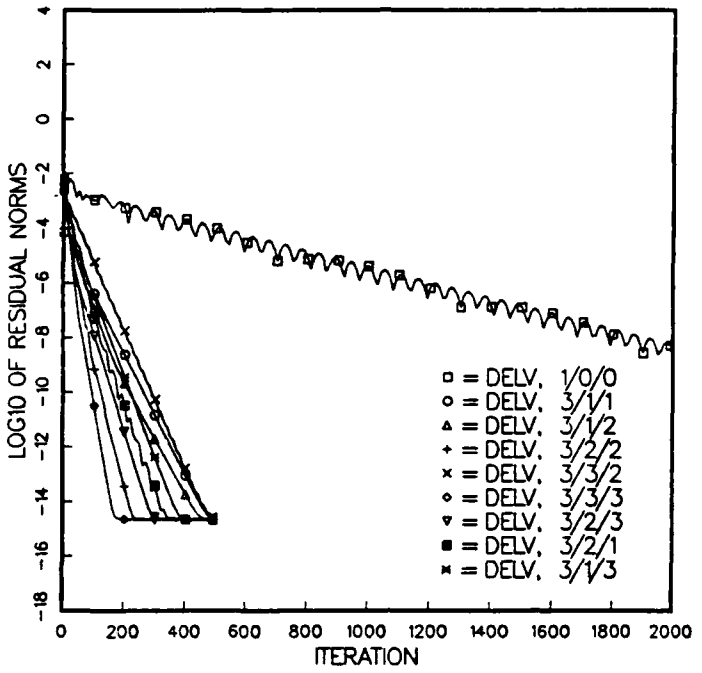


Figure 8.14. Convergence history for V flux, 21-by-21 grid, stationary, viscous flow, Re = 100, multiple grid runs

Table 8.8
Comparison of Convergence Properties for Multiple Grid Runs: Stationary Viscous Flow, 21-by-21 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 100

# Grids	# Fine Sweeps	# Coarse Sweeps	# Iterations	Relative Time to Con Tolerance	Speed Up Factor
1	N/A	N/A	1500	38.8	1.00
3	1	1	100	9.9	3.91
3	2	1	100	12.4	3.14
3	2	2	50	7.0	5.55
3	2	3	60	9.4	4.14
3	3	2	220	36.1	1.07
3	3	3	50	9.0	4.30
3	1	2	80	9.2	4.09
3	1	3	100	13.2	2.95

correct $\partial p/\partial x$ term; the $\partial p/\partial y$ term, which should be zero, was 10^{-7} (or smaller) for these runs as well.

The results of this section seem to agree with the hypothesis set forth in the previous section that either 1 or 2 relaxation sweeps on the finest grid are generally optimum when coupled with an approximately equal number of coarse-grid relaxation sweeps. While this rule cannot be definitively stated, it does stand as an excellent rule-of-thumb for multigrid runs. It can be stated that use of this rule, while possibly failing to provide the optimal convergence acceleration, would provide reliable and excellent convergence acceleration in its own right.

The maximum number of grids allowable/1 finest-grid relaxation sweep/maximum number of coarse-grid relaxation sweeps scenario (3/1/3) had the second worst performance among the combinations tested. This result is directly in keeping with the trend postulated in previous sections. Such a combination should not be used with the numerical scheme documented herein and the initial conditions employed.

8.3.2 Results for the 41-by-41 grid, Re = 100

For this scenario, the three best sweep combinations for the 3-grid and 4-grid Couette flow simulations on the 41-by-41 grid (section 8.2.3 above) were run. While use of the 3-grid combinations does not conform with the suggested utilization of the maximum number of grids allowable for a given problem (4 being maximum for this scenario), their use only provides additional investigation of this suggestion for a new physical problem. Results for these runs, and the fine grid-only run, are shown in Tables 8.9 and 8.10, and in Figures 8.15-8.17.

Table 8.9 again bears out the ability of both the fine grid-only and the multigrid runs to produce very accurate answers. The analytical $\partial p/\partial x$ for this scenario, 0.0020, was achieved for all the runs in this section. The 3-grid combinations all performed quite well. However, as

Table 8.9
Summary of Convergence Norms for
Multiple Grid Runs: Stationary, Viscous Flow,
41-by-41 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 100

		Average			Maximum		
Run	$\nabla \cdot u$	$\partial p/\partial t$	ΔU	ΔV	$\partial p/\partial t$	ΔU	ΔV
1/0/0	7.6e-08	6.5e-11	2.3e-10	6.1e-11	1.4e-09	5.1e-10	2.7e-10
4/1/1	3.1e-12	1.1e-15	6.3e-15	2.8e-15	1.0e-13	1.6e-14	1.8e-14
4/1/2	1.7e-09	6.8e-12	1.9e-12	1.6e-12	1.1e-10	5.5e-12	6.2e-12
4/2/1	3.5e-12	1.1e-15	6.6e-15	2.9e-15	7.3e-14	1.7e-14	1.9e-14
3/1/2	1.1e-10	1.2e-12	9.4e-14	4.4e-13	1.6e-11	2.6e-13	1.1e-12
3/1/3	2.0e-10	3.3e-12	7.9e-12	6.4e-12	2.9e-11	1.9e-11	2.6e-11
3/2/3	2.7e-12	2.1e-15	1.9e-14	8.2e-15	1.6e-13	4.7e-14	5.3e-14

Table 8.10
Comparison of Convergence Properties for
Multiple Grid Runs: Stationary Viscous Flow,
41-by-41 Grid Problem
Convergence Tolerance - 0.0001
Reynolds Number = 100

# Grids	# Fine Sweeps	# Coarse Sweeps	# Iterations	Relative Time to Con. Tolerance	Speed Up Factor
1	N/A	N/A	4500	411.2	1.00
4	1	1	110	42.2	9.73
4	1	2	200	90.8	4.53
4	2	1	105	49.4	8.33
3	1	2	180	75.6	5.44
3	1	3	135	65.1	6.32
3	2	3	150	80.7	5.10

shown in Table 8.10, the 4/1/1 and 4/2/1 combinations were still superior. This result, then, removes all question about the need to use the maximum number of grids allowable for a given finest-grid resolution to maximize convergence performance.

The 4/1/2 combination did not perform nearly as efficiently as the 4/1/1 or 4/2/1 combinations did. In fact, the 4/1/2 run required an under-relaxation coefficient of 0.35 as compared with 0.9 for the other

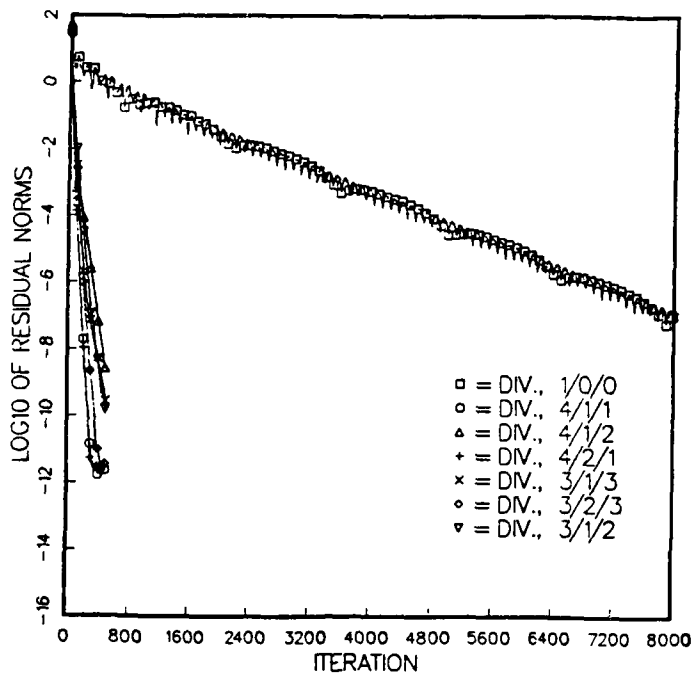


Figure 8.15a. Fine-grid-only and multigrid simulations

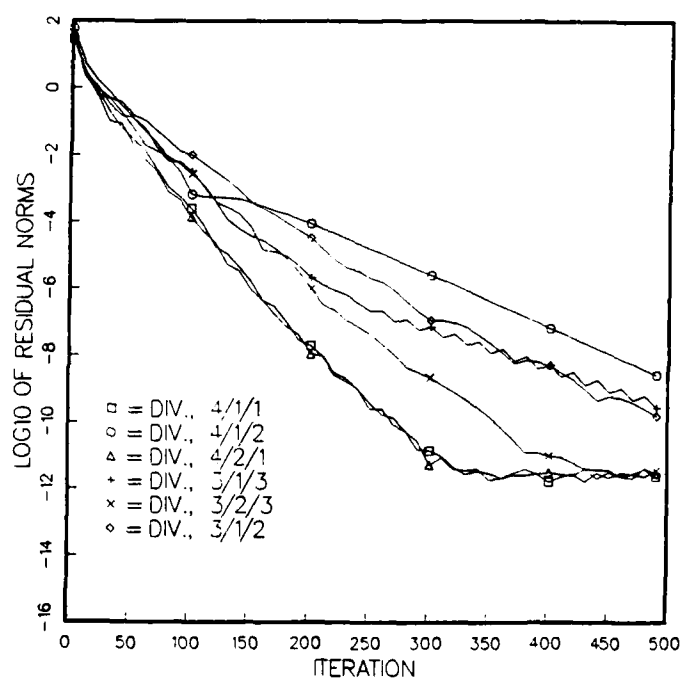


Figure 8.15b. Inset of 8.15a showing multigrid simulation

Figure 8.15. Convergence history for divergence of velocity, 41-by-41 grid, stationary, viscous flow, Re = 100, multiple grid runs

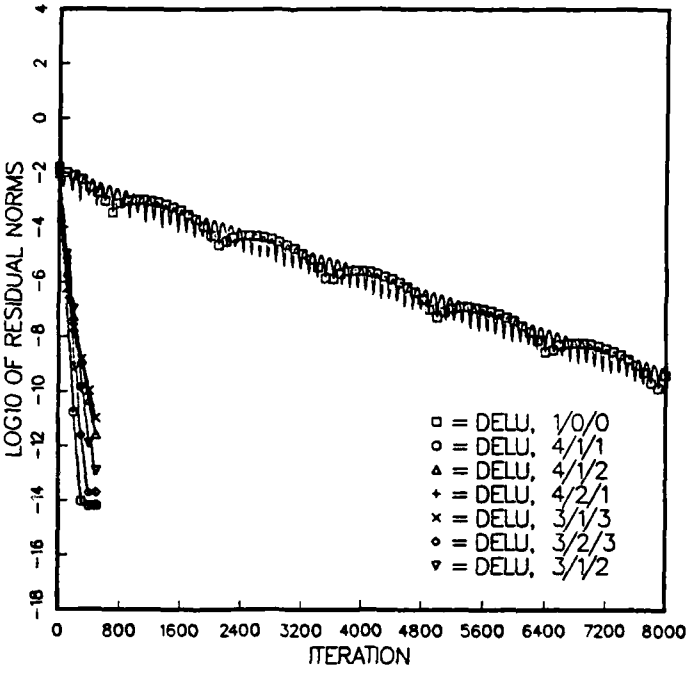


Figure 8.16a. Fine-grid-only and multigrid simulations

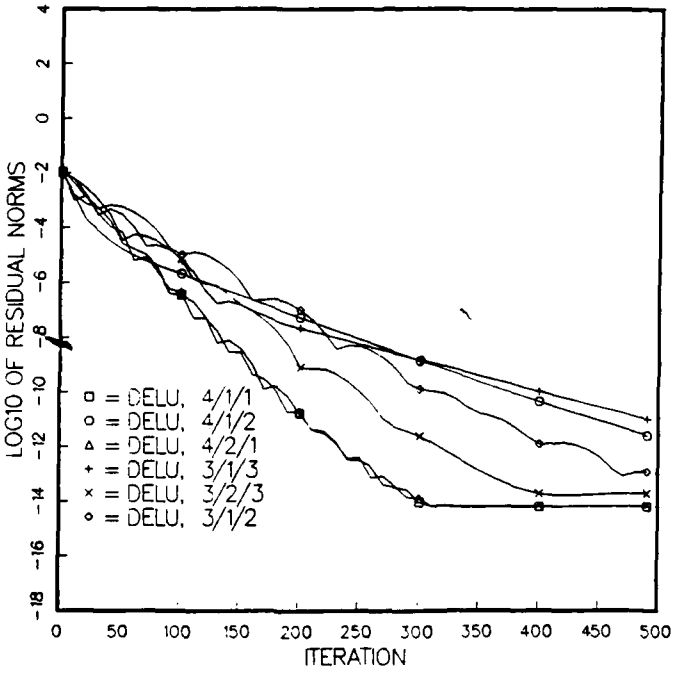


Figure 8.16b. Inset of 8.16a showing multigrid simulations.

Figure 8.16 Convergence history for U flux, 41-by-41 grid, stationary, viscous flow, Re = 100, multiple grid runs

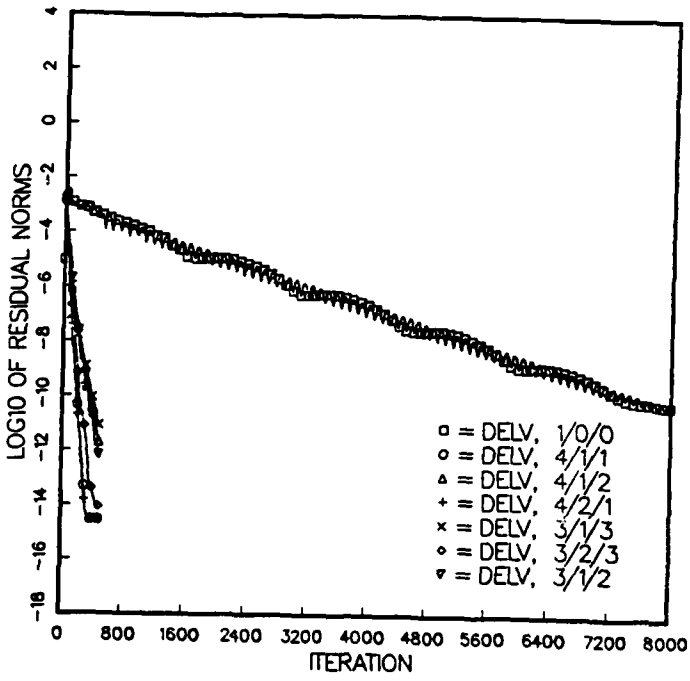


Figure 8.17a. Fine-grid-only and multigrid simulations

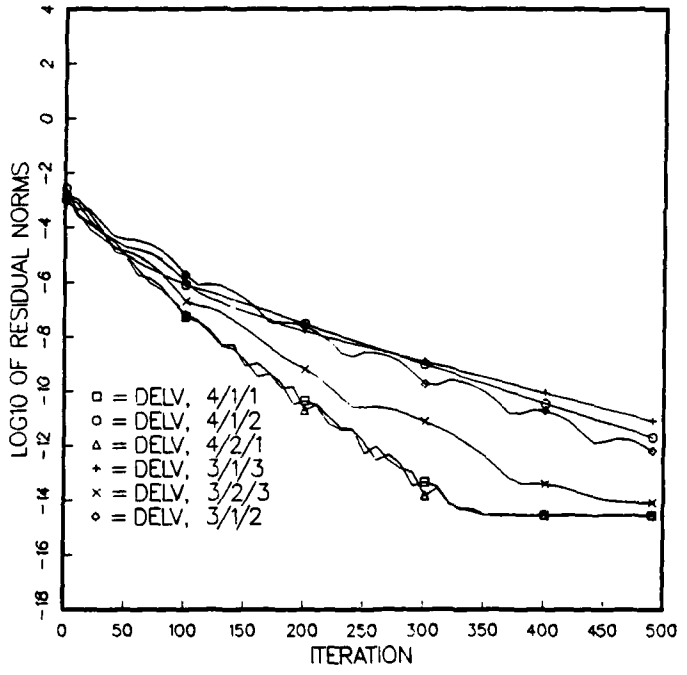


Figure 8.17b. Inset of 8.17a showing multigrid simulations

Figure 8.17. Convergence history for V flux, 41-by-41 grid, stationary, viscous flow, Re = 100, multiple grid runs

two 4-grid runs. Some effort was expended to optimize the value of this coefficient for the 4/1/2 run. This result, coupled with previous trends, suggests that the most reliable combination to use for excellent, if not optimal, multigrid convergence acceleration is the maximum number of allowable grids/one or two finest-grid relaxation sweeps/one coarse-grid relaxation sweep combination. It is this suggested rule which will be used as a basis for simulating more rigorous model test cases in the future. As a final note to this chapter, it appears that good, though not excellent, convergence acceleration can also be had through the use of fewer than the maximum number of grids when coupled with 1 or two finest-grid relaxation sweeps and 2 or 3 coarse-grid relaxation sweeps, respectively. These combinations, however, will be less efficient than the suggested combinations' results.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS

9.1 CHAPTER SUMMARY

This chapter provides a series of conclusions on the efficacy of the numerical scheme presented herein for the solution of the incompressible Navier-Stokes equations. The prospectus for using this scheme for future three-dimensional calculations for approach flows to hydraulic structures is discussed. Comments are included on the software design of the algorithm, its interface with the CYBER-205, and possible changes to the study results through use of a different supercomputer. Specific discussion of the components of the code, including an overview of its formulation, subroutines, memory requirements, etc., are provided in Appendix A.

Following the discussion and conclusions section, recommendations are made for additional research needed on the numerical scheme for both two-dimensional and three-dimensional applications.

9.2 DISCUSSION AND CONCLUSIONS

A numerical scheme has been described that is used to integrate the two-dimensional Navier-Stokes equations for steady-state, homogenous incompressible flow on a staggered grid. The approach has been limited to flows of low Froude number, thereby allowing the use of a rigid-lid approach. This flow is analogous to that expected in the approaches to many hydraulic structures. The explicit predictor-corrector finite volume relaxation scheme of MacCormack (1969) is used. It is coupled

with the pseudo-compressibility methodology of Chorin (1967), thus negating the need to solve a Poisson equation relating the pressure and flux fields. Results from the simulation of four model case studies are presented which show the efficacy of the presented numerical method.

A multigrid convergence acceleration technique patterned after Brandt (1984) and Jameson and Yoon (1986) was coupled with the basic relaxation scheme to enhance code efficiency. Results for the four model test cases conclusively showed the validity and attractiveness of the multigrid approach. The flow fields generated through inclusion of multigrid were as accurate as the finest grid-only calculations. These model test cases were generally observed to converge to a pre-assigned convergence tolerance in from 3 to 12 times faster than the finest grid-only solutions. Although the multigrid scheme did require more computer coding and produced up to a nine-fold increase in time per cycle compared to the fine grid-only calculations, it converged to the correct solution in many fewer iterations than did the non-multigrid runs. It appears potentially beneficial to add some amount of refinement to the finest grid so that more coarse grids could be effectively used. This suggests that it might be possible to increase convergence acceleration while increasing finest-grid refinement.

The optimal multigrid setup included utilizing the maximum number of total grids allowable given the resolution on the finest grid. The convergence was most faithfully accelerated by the use of one or two relaxation sweeps on the finest grid in concert with the use of one or two relaxation sweeps on each of the coarser grids. Sometimes extreme under-relaxation of the β coefficient or the multiplier on the time step produced faster convergence. However, this required a potentially

laborious sensitivity analysis. This need to "calibrate" the model is probably of questionable worth. Indeed, the combinations encompassed by the suggested rule above were already very near the optimum combination simulated for a given case.

Incorporation of the multigrid scheme into the basic relaxation procedure was often tedious. The results confirm the utility of the multigrid approach. Unfortunately, documentation of the multigrid approach is often too general or esoteric to be of any use for many numerical developers.

During the simulation of the Couette flow with a pressure gradient case, it was found that a second-order boundary condition for velocity was required to produce the correct shear stresses on the no-slip boundaries. Such a boundary condition was developed and used throughout simulations making up this report.

With a single exception, no effort was made to globally, or generally even locally, optimize the value of the β coefficient in the pseudo-compressibility calculations. It is possible that some increased convergence performance could have been obtained had this been done. However, this increased performance could have been offset by the resources required to optimize β for each of the conditions simulated. Thus, all runs utilized the largest β value that would provide reasonable flow field results without additional investigation.

The initial conditions used herein, that of a zero pressure and velocity field coupled with known fluxes on the boundaries, produced large initial continuity violations for all cases simulated. This was purposeful in that it was desired to show for every case that the model could recover from very adverse initial conditions and produce accurate

results. The use of differing initial conditions could have produced simulation results and trends different from those reported herein. However, it is advantageous to have a numerical scheme that is stable enough to basically start itself up rather than having to be feed a favorable initial condition set for adequate convergence. The use of the zero initial conditions with the grid/relaxation sweep combinations recommended above produces stable and accurate results relatively efficiently.

Some combinations of grid/relaxation sweep were slightly unstable and required under-relaxation. A few runs even appeared totally unstable. This problem may be associated with the initial conditions chosen. However, it is equally likely that the particular multigrid scheme used herein produced errors on the finest grid that it could not easily damp. Jameson and Yoon (1986) report a similar concern that led them to incorporate a dampening term in their scheme. No effort to develop a dampening methodology was made; however, it stands as a possible research need.

At the beginning of this report it was stated that the pseudo-compressibility method was to be used, rather than a Poisson solution, because the latter was iterative while the former allowed direct solution. It appears from the results presented in this report that the pseudo-compressibility method produces excellent results. Still, as vividly shown for the nonorthogonal grid case study, the skewness introduced resulted in a much-prolonged convergence of the basic relaxation scheme when compared to the analogous orthogonal case study. The use of a Poisson solver that is vectorizable, such as that used by Bernard (1988), would possibly be of worth for nonorthogonal problems.

Additionally, personal discussions with several numerical researchers indicated that the given convergence tolerances used for the model cases presented herein were usually reached in less than 500 iterations for schemes similar to that investigated. The lone general difference between their schemes and the one presented was their solution of a Poisson equation for pressure rather than the present use of pseudo-compressibility. Thus, the Poisson solution may be more rigorous than the pseudo-compressibility methodology in removing mass violations throughout the field for both orthogonal and nonorthogonal grids. Investigation of this possibility will commence in the very near future.

It appears that all indicators are positive for use of the general numerical scheme presented in three-dimensional calculations of approach flows to hydraulic structures. The scheme is accurate and relatively efficient for the model case studies. Further, it is relatively easy to code and, being explicit, to trouble-shoot. The results presented stand as a good foundation in support of further development of this scheme for steady-state, three-dimensional incompressible simulations of low-Froude number flows such as those often approaching hydraulic structures. The final recommendation of the use of pseudo-compressibility is qualified only in that more efficient solutions may be available through the implementation of a Poisson solver for pressure.

In the area of code development and formulation, no effort was made to customize this code for use on the CYBER-205. Some 205-specific vector functions were used, such as those computing the minimum and maximum of a given multi-dimensional array (Q8SMIN and Q8SMAX, respectively). Additional use of explicit vector syntax was made in

several places to initialize arrays. This was, however, the general extent of explicit vector functionality in this code. The result of this was a code that was relatively portable but slower than optimum on the CYBER-205. Portability was deemed the more important of these two, however, because of the strong likelihood of code usage on other mainframe supercomputers, such as a CRAY-2 or ETA-10, by the US Army Engineer Waterways Experiment Station. Thus, some performance was lost on the CYBER-205 as a result. As an example, the step size (stride) for the DO-loops controlling the coarse-grid relaxation sweeps was always an integer multiple of two. The CYBER-205 does not vectorize DO-loops that do not have unit (1) strides. Thus, none of the major loops in the coarse-grid relaxation sweep portion of this code vectorized. These loops would vectorize on a CRAY-2 or an ETA-10, however. In addition, the use of vector gathers and scatters on the CYBER-205 would alleviate this problem as well. These functions are specific to the 205 and would, therefore, negate the portability of the code. Thus, the performance exhibited by the code as presented would undoubtedly represent a lower bound when compared to its use on some other supercomputers.

The code (as discussed in Appendix A) was constructed with both portability and machine architecture in mind. Whenever possible both were placated. Approximately four out of every 10 DO-loops in the code vectorized on the CYBER-205. Most that did not either had non-unit DO-loop strides by requirement (such as the coarse-grid relaxation coding as discussed above), or were in the output portions of the code. Several of the non-vectorizing loops had diagnostics within them that were left in place throughout the simulations. This was necessary

because there was no a priori knowledge of when the diagnostics would be needed. Their presence had little impact on the overall run times for the scheme in any case.

As a final note in this section, it is the opinion of the author that the use of model test cases having known analytical or established solutions was of great benefit. Many code inadequacies and modifications were discovered directly from comparison of the numerical simulations with known values. The relative simplicity of the test cases from a fluid mechanics point-of-view is considered a large plus in that the features of the simulated flows, and the causes of those features, were well understood. Further, the combination of test cases had all the components of the more sophisticated problems, thereby allowing a thorough checkout of the various coding sequences. It is truly believed that test cases of these types should be attacked and researched thoroughly prior to "real-world" calculations in order to build the understanding and confidence needed to tackle these latter problems.

9.3 RECOMMENDATIONS FOR FUTURE WORK

It has become apparent that there are many more areas to investigate relative to the numerical scheme presented. Specific recommendations are presented below.

Under-relaxation of the changes in coarse-grid pressure and fluxes during prolongation should be investigated. This could aid in the mitigation of error introduction from the coarse grids to the finest solution.

There is very little in the literature concerning the nature of MacCormack's predictor-corrector scheme for three-dimensional,

incompressible simulations. More investigation into this is needed.

More rigorous investigation of the presented numerical scheme in two dimensions is warranted as well. In particular, a driven cavity or backstep case should be simulated. This will be done as a direct follow-on to this report.

Investigation of additional boundary conditions should be conducted. In particular, the fluid interaction with a no-slip boundary should be explored more fully, and boundary conditions compatible with staggered-grid schemes developed as needed.

Much more general research into the nature of multigrid solvers relative to incompressible flow simulation is needed. Very little is available in the literature on this subject. Information for both two-dimensional and three-dimensional incompressible simulations would be beneficial. It would also be helpful if documentation of multigrid mechanics, rather than theory, were presented.

The smoothing properties of the presented scheme need to be evaluated in some detail. Initial cursory evaluation suggested the scheme to be a poor smoother. This would allow an examination of more optimal convergence acceleration methodologies. It should be stressed that the aim of the research presented herein was not to develop the most efficient multigrid solver available; rather, it was to examine the potential for speeding up the convergence of the presented relaxation scheme that itself was known to be accurate for incompressible flow simulation. Thus, future modifications of the numerical methodology presented should be examined for enhanced smoothing and convergence performance.

Finally, the use of a Poisson pressure solver rather than pseudo-compressibility is in order. This, too, is expected to be done in the near-future.

REFERENCES

Anderson, D. A., Tannehill, J. C., and Pletcher, E. H. 1984. Computational Fluid Dynamics and Heat Transfer, McGraw-Hill, New York.

Aref, H. 1986. "The Numerical Experiment in Fluid Mechanics," Journal of Fluid Mechanics, Vol. 173, pp. 15-41.

Aziz, K., and Hellums, J. D. 1967. "Numerical Solution of the Three-Dimensional Equations of Motion for Laminar Natural Convection," Physics of Fluids, Vol. 10, No. 2, pp. 314-324.

Baker, A. J. 1983. Finite Element Fluid Mechanics, McGraw-Hill, New York.

Beam, R. M., and Warming, R. F. 1976."An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form," Journal of Computational Physics, Vol. 22, pp. 87-110.

Bercovier, M., Berwald, G., and Pironneau, O. 1981. "A Finite Element Method for the Numerical Solution of Transient Inviscid Incompressible Flows," Computational Techniques in Transient and Turbulent Flow, Vol 2.

Bernard, R. S. 1986. "Discrete Solution of the Anelastic Equations for Mesoscale Modeling," Report GKSS 86/E/51, GKSS Forschungszentrum, Postfach 1160, 2054 Geesthacht, West Germany.

Bernard, R. S. 1987. "Skew Grids and Irrotational Flow," Proceedings of the Fifth Army Conference on Applied Mathematics and Computing, West Point, NY.

Bernard, R. S. 1988. "Explicit Numerical Algorithm for Modeling Incompressible Approach Flow; Hydraulics Laboratory Investigation," Technical Report REMR-HY-4, US Army Engineer Waterways Experiment Station, Vicksburg, MS.

Bernard, R. S., and Thompson, J. F. 1984. "Mass Conservation on Regular Grids for Incompressible Flow," AIAA Paper 84-1669, AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, CO.

Brandt, A. 1977. "Multi-Level Adaptive Solutions to Boundary-Value Problems," Mathematics of Computation, Vol. 31, No. 138, pp. 333-390.

Brandt, A. 1984. "Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics," Gesellschaft Fur Mathematik Und Datenverarbeitung Report Number 85, Bonn, FRG.

Briley, W. R., and McDonald, H. 1973. "Solution of the Three-Dimensional Compressible Navier-Stokes Equations by an Implicit Technique," United Aircraft Research Laboratories Report M911363-6.

Chang, J. L. C., and Kwak, D. 1984. "On the Method of Pseudo Compressibility for Numerically Solving Incompressible Flows," AIAA Paper 84-0252, AIAA 22nd Aerospace Sciences Meeting, Reno, NV.

Chang, J. L. C., Kwak, D., Dao, S. C., and Rosen, R. 1985. "A Three-Dimensional Incompressible Flow Simulation Method and Its Application to the Space Shuttle Main Engine, Part I - Laminar Flow," AIAA Paper 85-0175, AIAA 23rd Aerospace Sciences Meeting, Reno, NV.

Chen, R. C. 1982. "Development of Finite Analytic Method for Unsteady Three-Dimensional Navier-Stokes Equations," PhD Thesis, University of Iowa.

Chen, Y. S. 1986. "A Numerical Method for Three-Dimensional Incompressible Flows Using Nonorthogonal Body-Fitted Coordinate Systems," AIAA Paper 86-1654, AIAA/ASME/SAE/ASEE 22nd Joint Propulsion Conference, Huntsville, AL.

Chien, J. C., and Schetz, J. A. 1975. "Numerical Solution of the Three-Dimensional Navier-Stokes Equations with Applications to Channel Flows and a Buoyant Jet in a Cross Flow," Journal of Applied Mechanics, pp 575-579.

Chima, R. V. 1986. "Development of an Explicit Multigrid Algorithm for Quasi-Three-Dimensional Viscous Flows in Turbomachinery," NASA Technical Memorandum 87128, Lewis Research Center, Cleveland, OH.

Chima, R. V., and Johnson, G. M. 1983. "Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm," NASA Technical Memorandum 83376, Lewis Research Center, Cleveland, OH.

Choi, D., and Merkle, C. L. 1984. "Application of Time-Iterative Schemes to Incompressible Flow," AIAA Paper 84-1638, AIAA 17th Fluid Dynamics, Plasma Dynamics and Lasers Conference, Snowmass, CO.

Chorin, A. J. 1967. "A Numerical Method for Solving Incompressible Viscous Flow Problems," Journal of Computational Physics, Vol. 2, pp. 12-26, 1967.

Cooley, J. H., Jr. 1986. "A Vectorized Numerical Solution Algorithm for Three-Dimensional Incompressible Fluid Flow," Master's Thesis, Colorado State University, Fort Collins, CO.

Courant, R., Friedrichs, K. O., and Lewy, H. 1929. "Uber die Partiellen Differenzengleichurgen der Mathematischen Physik," Mathematische Annalen, Vol. 100, pp. 32-74 (English translation).

Ferziger, J. H. 1987. "Simulation of Incompressible Turbulent Flows," Journal of Computational Physics, Vol. 69, pp 1-48.

Fromm, J. E. 1963. "A Method for Computing Nonsteady Incompressible Viscous Fluid Flows," Los Alamos Science Laboratory Report LA-2910, Los Alamos, NM.

Fuchs, L., and Zhao, H. S. 1984. "Solution of Three-Dimensional Viscous Incompressible Flows by a Multi-Grid Method," International Journal for Numerical Methods in Fluids, Vol. 4, pp. 539-555.

Gresho, P. M., Chan, S. T. K., Lee, R. L., and Upson, C. O. 1981. "Solution of the Time-Dependent Incompressible Navier-Stokes Equations Via FEM," Numerical Methods in Laminar and Turbulent Flow, Proceedings of the Second International Conference, Venice.

Gunzburger, M. D., Nicolaides, R. A., and Liu, C. H. 1983. "New Discretization and Solution Techniques for Incompressible Viscous Flow Problems," NASA Contractor Report 172196, Langley Research Center, Hampton, VA.

Harlow, F. H., and Welch, J. E. (1965). "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," Physics of Fluids, Vol. 8, No. 12, pp. 2182-2189.

Hiraski, G. J., and Hellums, J. D. 1970. "Boundary Conditions on the Vector Potentials in Viscous Three-Dimensional Hydrodynamics," Quarterly of Applied Mathematics, Vol. 28, No. 2, pp. 293-296.

Hirt, C. W., and Cook, J. L. 1972. "Calculating Three-Dimensional Flows Around Structures and Over Rough Terrain," Journal of Computational Physics, Vol. 10, pp. 324-340.

Horiuti, K. 1985. "Large Eddy Simulation of Turbulent Channel Flow," Institute of Industrial Science, Annual Report of Group Research Activity on Numerical Simulation of Turbulent Flows, Number 1, University of Tokyo.

Jameson, A., Schmidt, W., and Turkel, E. 1981. "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, CA.

Jameson, A., and Yoon, S. 1986. "Multigrid Solution of the Euler Equations Using Implicit Schemes," AIAA Journal, Vol. 24, No. 11.

Johnson, B. J. 1981. "A Review of Numerical Reservoir Hydrodynamic Modeling," Technical Report E-81-2, US Army Engineer Waterways Experiment Station, Vicksburg, MS.

Johnson, G. M. 1982. "Multiple-Grid Acceleration of Lax-Wendroff Algorithms," NASA Technical Memorandum 82843, Lewis Research Center, Cleveland, OH.

Johnson, G. M. 1983. "Multiple-Grid Convergence Acceleration of Viscous and Inviscid Flow Computations," NASA Technical Memorandum 83361, Lewis Research Center, Cleveland, OH.

Kim, J., and Moin, P. 1984. "Application of a Fractional-Step Method to Incompressible Navier-Stokes Equation," NASA Technical Memorandum 85898, Ames Research Center, Moffett Field, CA.

Kirtley, K. R., Warfield, M., and Lakshminarayana, B. 1986. "A Comparison of Computational Methods for Three-Dimensional Turbulent Turbomachinery Flows," AIAA Paper 86-1599, AIAA/ASME/SAE,ASEE 22nd Joint Propulsion Conference, Huntsville, AL.

Kreyszig, E. 1979. Advanced Engineering Mathematics, Fourth Edition, Wiley and Sons, New York.

Kwak, D., Chang, J. L. C., Shanks, S. P., and Chakravarthy, S. R. 1984. "An Incompressible Navier-Stokes Flow Solver in Three-Dimensional Curvilinear Coordinate System Using Primitive Variables," AIAA Paper 84-0253, AIAA 22nd Aerospace Sciences Meeting, Reno, NV.

Laval, H. 1981. "CONVEC: A Computer Program for Transient Incompressible Fluid Flow Based on Quadratic Finite Elements; Part I: Theoretical Aspects," Report EUR 7427/1 EN, Commission of the European Communities, Brussels.

Lustman, L. 1984. "A Review of Spectral Methods," ICASE Report 84-59, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA.

MacCormack, R. W. 1969. "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, AIAA Hypervelocity Impact Conference, Cincinnati, OH.

MacCormack, R. W. 1985. "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA Paper 85-0032, AIAA 23rd Aerospace Sciences Meeting, Reno, NV.

Martinelli, L., Jameson, A., and Grasso, F. 1986. "A Multigrid Method for the Navier-Stokes Equations," AIAA Paper 86-0208, AIAA 24th Aerospace Sciences Meeting, Reno, NV.

Mastin, C. W., and Thompson, J. F. 1978. "Three-Dimensional Body-Fitted Coordinate Systems for Numerical Solution of the Navier-Stokes Equations," AIAA Paper 78-1147, AIAA 11th Fluid and Plasma Dynamics Conference, Seattle, WA.

McCormick, S. F. 1982. "An Algebraic Interpretation of Multigrid Methods," SIAM Journal, Vol. 19, No. 3.

McCormick, S. 1985. "Special Issue: Second Copper Mountain Conference on Multigrid Methods," Applied Mathematics and Computation, Vol. 19, Nos. 1-4.

McCormick, S., and Trottenberg, U. 1983. "Special Issue: Multigrid Methods," Applied Mathematics and Computation, Vol. 13, Nos. 3 and 4.

Moin, P., and Kim, J. 1982. "Numerical Investigation of Turbulent Channel Flow," Journal of Fluid Mechanics, Vol. 118, pp. 341-377.

Moitra, A. 1982. "An Implicit Solution of the Three-Dimensional Navier-Stokes Equations for an Airfoil Spanning a Wind Tunnel," PhD Dissertation, Mississippi State University.

Moitra, A., Turkel, E., and Kumar, A. 1986. "Application of a Runge-Kutta Scheme for High-Speed Inviscid Internal Flows," AIAA Paper 86-0104, AIAA 24th Aerospace Sciences Meeting, Reno, NV.

Neely, G. M., and Claus, R. W. 1985. "Accelerated Convergence for Incompressible Flow Calculations," NASA Technical Memorandum 86863, Lewis Research Center, Cleveland, OH.

Ni, R. H. 1982. "A Multiple-Grid Scheme for Solving the Euler Equations," AIAA Journal, Vol. 20, No. 11.

Orszag, S. A., and Israeli, M. 1974. "Numerical Simulation of Viscous Incompressible Flows," Annual Review of Fluid Mechanics, Vol. 6.

Oswalt, N. R. 1988. Unpublished information and personal communication, US Army Engineer Waterways Experiment Station, Vicksburg, MS.

Patankar, S. V. 1981. "A Calculation Procedure for Two-Dimensional Elliptic Situations," Numerical Heat Transfer, Vol. 4, pp. 409-425.

Patankar, S. V., and Spalding, D. B. 1972. "A Calculation Procedure for Heat, Mass, and Momentum Transfer in Three-Dimensional Parabolic Flows," International Journal of Heat and Mass Transfer, Vol. 15, pp. 1787-1806.

Patel, N. R., and Thompson, J. F. 1984. "A Vectorized Solution for Incompressible Flow," AIAA Paper 84-1534, AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, CO.

Peyret, R., and Taylor, T. D. 1983. Computational Methods for Fluid Flow, Springer-Verlag, New York.

Redhed, D. D., Chen, A. W., and Hotovy, S. G. 1979. "New Approach to the 3-D Transonic Flow Analysis Using the STAR 100 Computer," AIAA Journal, Vol. 17, pp. 98-99.

Reizes, J. A., Leonardi, E., and de Vahl Davis, G. 1984. "Problems with Derived Variable Methods for the Numerical Solution of Three-Dimensional Flows," Computational Techniques and Applications, Elsevier Science Publishers.

Rhie, C. M. 1986. "A Pressure Based Navier-Stokes Solver Using the Multigrid Method," AIAA Paper 86-0207, AIAA 24th Aerospace Sciences Meeting, Reno, NV.

Roache, P. J. 1972. Computational Fluid Dynamics, Hermosa, Albuquerque, NM.

Rogers, S. E., Kwak, D., and Kaul, U. K. 1986. "A Numerical Study of Three-Dimensional Incompressible Flow Around Multiple Posts," AIAA Paper 86-0353, AIAA 24th Aerospace Sciences Meeting, Reno, NV.

Rosenfeld, M., and Israeli, M. 1987. "Numerical Solution of Incompressible Flows by a Marching Multigrid Nonlinear Method," AIAA Journal, Vol. 25, No. 5.

Shang, J. S., Buning, P. G., Hankey, W. L., and Wirth, M. C. 1980. "Performance of Vectorized Three-Dimensional Navier-Stokes Code on the CRAY-1 Computer," AIAA Journal, Vol. 18, pp. 1073-1079.

Smith, R. E., and Pitts, J. I. 1979. "The Solution of the Three-Dimensional Compressible Navier-Stokes Equations on a Vector Computer," Third IMACS Symposium on Computer Methods for Partial Differential Equations, Lehigh University, PA.

Soh, W. Y., and Berger, S. A. 1984. "Laminar Entrance Flow in a Curved Pipe," Journal of Fluid Mechanics, Vol. 148, pp. 109-135.

Steger, J. L. 1978. "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries," AIAA Journal, Vol. 16, No. 7, pp. 679-686.

Streeter, V. L., and Wylie, E. B. 1985. Fluid Mechanics, 8th Edition, pp. 191-196, McGraw-Hill, New York.

Stuben, K., and Trottenberg, U. 1982. "Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Application," Lecture Notes in Mathematics, 960, Springer-Verlag, Berlin.

Thompson, J. F. 1984. "Grid Generation Techniques in Computational Fluid Dynamics," AIAA Journal, Vol. 22, No. 11.

Vanka, S. P., and Misegades, K. P. 1987. "Vectorized Multigrid Fluid Flow Calculations on a Cray X-MP/48," International Journal for Numerical Methods in Fluids, Vol. 7, pp. 635-648.

Vinokur, M. 1976. "The Relation of Finite Element and Finite Difference Methods," NASA Contractor Report CR-2764, Ames Research Center.

Yeh, G. T. 1981. "Numerical Solutions of Navier-Stokes Equations with an Integrated Compartment Method," International Journal for Numerical Methods in Fluids, Vol. 1, pp. 207-221.

APPENDIX A

DESCRIPTION OF COMPUTATIONAL SOFTWARE

A.1 INTRODUCTION

This appendix provides an overview of the computerization of the numerical algorithm described in the main text. This overview includes synoptic discussion of each of the code's subroutines; a listing of the input to the code; and a delineation of the output from each simulation run. The appendix begins with a discussion of the basic formulation of the computer code.

A.2 BASIC COMPUTER FORMULATION

The numerical algorithm presented in the main text was developed using the FORTRAN 77 programming language; it was executed on a CYBER-205 that had two vector pipelines. In order to maximize code portability, few departures from standard FORTRAN 77 were made. A notable exception to this was the use of namelists for specification of input and output parameters. Additionally, explicit CYBER-205 vector syntax and intrinsic functions were used in some cases. These were usually limited in use, however, to cases of array initialization and to searches of two and three-dimensional arrays for minima and maxima. No vector gathers and scatters were used.

The code was developed in a modular fashion. A main routine was developed that housed calls to the grid setup subroutines and housed the master code iteration loop. Within this loop, the basic computational subroutines were called such as the MAC subroutine that employed the

basic relaxation scheme, the multigrid scheme, and made calls for updated boundary conditions. Each of these subroutines is summarized in a subsequent section of this appendix.

Note is the use of parameter statements throughout the code. This allowed a relative conservation of computer memory during the load and execution phase. As an example, the simulation of a given model test case on a 21-by-21 grid required only 395 thousand words of central memory to load and execute. An analogous 41-by-41 grid simulation required 723 thousand words.

In terms of run times, a typical simulation of a Couette flow test case on a 21-by-21 grid with no multigrid required 109 system billing units (this is a CYBER-205 internal pricing measure; 61.5 seconds is the analogous central processing time) to run 2000 iterations including setup and output. Given processing costs of $500 per system billing unit at Colorado State University, this run cost was approximately $15. The addition of multigrid (with 3 levels of grids) to this problem required 119 system billing units to run 500 iterations including setup and output. This translated to a cost of approximately $17. A typical simulation of the same Couette flow test case on a 41-by-41 grid with no multigrid had a cost of approximately $29.

A review of the DO-loops in the code showed that approximately 38 percent of the loops in the code vectorized on the CYBER-205 using safe vectorization. The primary reasons for the failure for vectorization were: (a) use of non-unit stride for the loops controlling the coarse-grid operations within the multigrid algorithm; and, (b) use of loops as a part of a write sequence during intermediate or final output creation. This level of vectorization obviously could have been enhanced on the

CYBER-205 through the use of vector gathers and scatters; however, such use would have reduced greatly the portability of the code.

A.3 SYNOPSIS OF SUBROUTINE COMPONENTS

This section provides an overview of the interworkings of the individual subroutines within the code. The routines are, following the main program, listed in alphabetical order. Only a short overview is provided for each routine. Detailed information on several components of the code, such as relaxation, restriction, or prolongation, is provided in the main body of this report.

A.3.1 MAIN Routine

This routine calls the main setup routines GRID, JACOB, INITIAL, TIME, and RHS initially. The allowable pseudo-compressibility coefficients are also computed for the finest grid only. Then, for a user specified number of iterations (or multigrid cycles), the routine calls the MAC subroutine that houses the relaxation and multigrid algorithms. At the end of each of these iterations, convergence norms, related to the divergence of velocity and the differences in the computed fluxes and their analytical solutions, are computed. If the solution has converged, subroutine OUTPUT is called to provide summary information and data file creation. If the solution has not converged, the iteration process is continued. If the solution fails to converge in the alloted number of iterations, the intermediate solution is presented by subroutine OUTPUT with a message that the solution failed to converge. The MAIN routine also provides times of execution of the main computational loop (i.e., central processing time for the performance of N multigrid iterations while neglecting the time required for setup and output) for diagnostic use later.

A.3.2 Subroutine BCC

This routine updates the pressure change values residing computationally just beyond the physical boundaries on each of the coarse grids. The routine extrapolates field values normal to a given boundary to obtain the value just beyond. The routine calls no other subroutines and is, itself, called from the multigrid section of the MAC subroutine.

A.3.3 Subroutine BCCV

This routine updates the u and v-velocity component values residing computationally just beyond the physical boundaries for each of the coarse grids. The routine uses a second-order extrapolation for both slip and no-slip boundaries. The routine calls no other subroutines and is, itself, called from the multigrid section of the MAC subroutine.

A.3.4 Subroutine CORPRE

This routine computes the flux-centered pressure derivatives which are components of the momentum equations. These derivatives, which are centered separately about the U and V flux locations, are then used to compute the changes to these fluxes as a result of the imposed pressure gradient. The same routine, which operates on the finest grid only, is used to compute the effects of the gradient of pressure changes on the respective fluxes. The subroutine calls no other routines and is called itself by the MAC routine.

A.3.5 Subroutine CPGRAD

This subroutine computes the same information as subroutine CORPRE, except that its calculations are made on the coarse grids. However, in contrast to CORPRE, CPGRAD does not use the pressure calculations to modify the existing coarse-grid fluxes. Rather, the pressure information is then restricted from the present coarse grid to a yet-coarser

one for later use. The routine is called from the multigrid section of subroutine MAC and calls no routines itself.

A.3.6 Subroutine FPGRAD

This routine performs the exact same task as subroutine CPGRAD on the finest, rather than coarser, grid. The routine is called from the multigrid section of subroutine MAC and calls no routines itself.

A.3.7 Subroutine GRID

This routine computes the x and y locations of each of the node points that make up a given numerical mesh. Generation of grids for 4 specific model test geometries (straight channel, half-cylinder in a channel, converging-diverging channel, arbitrarly-skewed channel), each having a user-specified number of points along the x and y axes, are possible. Other arbitrary geometries are not available for generation with this routine. The routine will also generate stretched grids for the given examples. The routine is called by the MAIN program and calls no subroutines itself.

A.3.8 Subroutine INITIAL

This subroutine provides the setup of initial conditions, boundary values, and problem type. Several model test cases can be simulated including two types of Couette flow; flow over a half-cylinder; flow through a converging-diverging channel; and flow in a driven cavity. The flow problem type is specified in the user input. Testing of the latter three physical cases has been limited to date. The routine also specifies which boundaries are slip and no-slip, and the constant boundary flux values, based on the test case being simulated. The routine, which calls no other subroutines, is called by the MAIN program.

A.3.9 Subroutine JACOB

Using the node-point information computed in the GRID subroutine, this routine computes the metrics and jacobians required in the transformation from cartesian coordinates to generalized curvilinear coordinates. The routine calls no other routines and is called by the MAIN program.

A.3.10 Subroutine MAC

This is the main computational subroutine in this code. It is called for every iteration or multigrid cycle from the MAIN program. Within MAC, the primary predictor-corrector relaxation scheme is used to compute modifications to the pressure and flux fields on both the finest and, if applicable, coarser grids. The shift condition, which is used within the predictor-corrector to modify the direction of its differencing, is set and maintained within MAC. Within the routine, the cell-centered velocity values are obtained from the existing flux field and the known grid metrics. Following use of the relaxation scheme, the fluxes and pressures are updated based on computed modifications to the velocity and pressure fields. If multigrid is being employed, the MAC routine computes coarse-grid metrics and jacobians, coarse allowable time steps and pseudo-compressiblity coefficients, and residuals. It then restricts these residuals and the existing-grid solution to a coarser grid, and completes the relaxation procedure on this coarser grid. This is done for a user-specified number of coarse grids. Upon completing these operations on the coarsest grid, the MAC routines performs a prolongation process to update the finest-grid solution. The MAC subroutine calls subroutines BCC, BCCV, PBC, and VBC for computation of updated boundary information. It calls subroutines CORPRE AND MULPRE

inorder to modify the existing fluxes on the finest and coarser grids, respectively, based on the effects of the pressure gradient resident on the respective grid. MAC calls CPGRAD and FPGRAD to provide for restriction of pressure gradient information from a finer grid to a coarser one. Additionally, MAC calls the following: subroutine RHS to compute the product of velocity and flux; subroutine RHSV to compute the viscous terms on the finest grid; subroutine RHSVC to compute the viscous terms on the coarser grids; and subroutine MACCO to compute the initial residuals projected on a coarse grid through relaxation using the just-restricted solution as an initial solution. This routine does the bulk of the work in this code.

A.3.11 Subroutine MACCO

This routine uses the predictor-corrector relaxation scheme to compute the residuals that would exist on a given coarse grid with the solution restricted from the just-finer grid acting as its initial solution. These residuals are then employed as part of the right-hand sides of the equations relaxed on the coarse grids within the multigrid section of the MAC subroutine. This routine, which is called by MAC, calls subroutines BCC, BCCV and RHSVC.

A.3.12 Subroutine MULPRE

This subroutine computes the modifications to the coarse-grid fluxes resulting from application of the pressure gradient on the given coarse grid. In this connection, it performs the exact same function as the CORPRE routine, except that the latter makes computations on the finest grid only. This routine calls no other routines and is called by MAC.

A.3.13 Subroutine NEWTON

This small routine computes the base functions used in the GRID subroutine for stretched coordinate generation using Newton-Raphson iteration. A convergence tolerance of 1.E-12 is used to ensure appropriate grid generation resolution. A maximum of 2000 iterations is allowed for convergence. This routine is called by subroutine GRID and calls no other routines.

A.3.14 Subroutine OUTPUT

This routine performs two specific tasks: (a) it outputs a hard-copy of several performance-related statistics, the final computed flow-field solution, the values of the streamfunction at all node locations, and run parameters; and (b) it generates up to 5 data files for ultimate plotting and analysis. These data files house the computed flow field, the streamfunction values, the x-y locations of the node points, the norms of the calculations at every 10th iteration or cycle, and the locations of those norms in the field. The routine is called by the MAIN program; it calls no subroutines.

A.3.15 Subroutine PBC

This routine updates the computational boundary values for pressure just beyond the physical boundaries on the finest grid only. It uses the same numerical boundary condition for these updates as used by subroutine BCC. This routine is called by subroutine MAC and calls no routines itself.

A.3.16 Subroutine PGDLOD

This routine uses bilinear interpolation to prolong pressure corrections from a given coarse grid to its just-finer counterpart. The routine is called by MAC, and it calls subroutine BCC to complete its calculations.

A.3.17 Subroutine RHS

This routine computes the product of a given flux and a given velocity (i.e., each of the four possible combinations of the U and V fluxes and the u and v velocities) for use in the relaxation of the momentum equations. The routine makes use of the shift condition set within subroutine MAC to relate these fluxes and velocities. The routine is called initially by the MAIN program, and by the MAC routine thereafter. RHS calls no other routines.

A.3.18 Subroutine RHSV

This routine computes the viscous terms for the Navier-Stokes equations based on the methodology shown in Appendix B for the finest grid only. Shear stresses along no-slip boundaries are also explicitly computed. This routine is called by subroutine MAC; it calls no routines.

A.3.19 Subroutine RHSVC

This subroutines performs exactly the same function as subroutine RHSV for the coarse grids only. It, too, is called by subroutine MAC and calls no routines itself.

A.3.20 Subroutine TIME

This routine computes the allowable time step based on a heuristic stability condition (as presented in Chapter 6) for the finest grid only. The routine then seeks out the minimum allowable time step within the field and assigns its value as the global time step for all cells within the computational domain. The computed allowable time step is multiplied by a user-specified factor, which ranges between 0 and 1, to allow some control over the impacts of non-linear instabilities. This

factor was generally set at 0.9. The routine is called by the MAIN routine. It calls no subroutines.

A.3.21 Subroutine VBC

This routine updates the computational velocity values in cells just beyond the physical boundaries using second-order relationships. These calculations are made by this routine on the finest grid only. Its coarse-grid counterpart is subroutine BCCV. This routine is called by subroutine MAC. It calls no routines itself.

A.4 INPUT SPECIFICATION

All of the input to this code is incorporated into a namelist entitled DATAIN. Within this namelist are logical, alphanumeric, and arithmetic variables which control all phases of the model test case selection; number of grid nodes; maximum number of iterations or multi-grid cycles; etc. The namelist was located within the main job control language used to execute the numerical modeling program. A short description of the items within the namelist DATAIN is given below. Four types of variables are within the namelist: floating-point (f); integer (i); logical (l); and character (c). The logical variables were either true for false.

BETAC - (f): Relaxation multiplier for pseudo-compressibility coefficient calculation. Value always between 0 and 1; usually 0.9.

CFL - (f): Relaxation multiplier for time step calculation. Value always between 0 and 1; usually 0.9.

DYFINE - (f): Size of smallest spatial step in vertical direction.

EPS - (f): Convergence tolerance for all norms. Value set to 1.E-16 whenever desire was to run a pre-specified number of iterations for generation of a convergence history.

ETYPE - (c): Six letter variable set to "NAVIER" to simulate the Navier-Stokes equations; set to "EULER " to simulate the Euler equations.

FIELDX - (f): Multiplies initial u-velocity boundary values to allow differing initial u-velocity conditions to be set in the field. Set to zero for all of the calculations reported in this report.

FIELDY - (f): Multiplies initial v-velocity boundary values to allow differing initial v-velocity conditions to be set in the field. Set to zero for all of the calculations reported in this report.

FNAME - (c): Up to 15 letter variable set to the file name to be assigned to the output data file housing the final pressure, velocity, and streamfunction values. This file name is placed in the first line of the file holding the flow data. The file is then transferred from the CYBER frontend at Colorado State University to a VAX 11/750 at the Waterways Experiment Station, Vicksburg, MS. This file name is used by the VAX as its new file name for the given file. This variable always started with the letters "QD" to designate it as pressure and velocity data (the Q vector housed these data values).

FNAME1 - (c): Same as FNAME, except that is was used to signify the data file housing the x and y geometry for each of the node points in the numerical mesh. Each of these files starts with the letters "XZ".

FNAME2 - (c): Same as FNAME, except that is was used to signify the data file housing the pressure and flux norms for each of the cells in the field. Each of these data files starts with the letters "QN".

FNAME3 - (c): Same as FNAME, except that is was used to signify the data file housing the norms for the divergence of velocity for each

of the cells in the field. Each of these files starts with the letters "DV".

ITMAX - (i): Maximum number of iteration or multigrid cycles to be simulated. This number can not exceed 10000 at present without re-dimensioning the code for an increased number.

IWSLP1 - (i): Initial i-location along a given section of a horizontal no-slip boundary which is to be designated as slip. If this number is greater than IWSLP2, the entire given boundary is designated as no-slip.

IWSLP2 - (i): Ending i-location along a given section of a horizontal no-slip boundary which is to be designated as slip.

IXBUMP - (i): Number of grid points along either the bottom or bottom and top axes that reside directly on the half-cylinder simulated in the converging-diverging channel and half-cylinder test cases. This option, and the next two associated with it, have not been fully tested as yet.

IXDOWN - (i): Number of grid points along either the bottom or bottom and top axes which reside in the grid region just downstream of the bump for the cases mentioned just above.

IXUP - (i): Number of grid points along either the bottom or bottom and top axes which reside in the grid region just upstream of the bump for cases mentioned just above.

JWSLP1 - (i): Initial j-location along a given section of a vertical no-slip boundary which is to be designated as slip. If this number is greater than JWSLP2, the entire given boundary is designated as no-slip.

JWSLP2 - (i): Ending j-location along a given section of a vertical no-slip boundary which is to be designated as slip.

NGRIDS - (i): Number of total grids, including the finest, to be simulated in a given multigrid scenario. The maximum number considered in this report was 4; the maximum the code is presently designed to consider is 5.

NPREIT - (i): Number of iterations to skip, after the first iteration, prior to printing intermediate or diagnostic information (such as velocity, pressure, and flux information). The minimum value of this variable was 1.

NPRNT - (i): Variable controlling output of convergence history information, allowed writing of data to output files every NPRNT iterations. This value was always 10 for the work reported in this report.

NSWEEPC - (i): Number of relaxation sweeps to make on each coarse grid per multigrid cycle.

NSWEEPF - (i): Number of relaxation sweeps to make on the finest grid per multigrid cycle.

OMEGAM - (f): Relaxation coefficient used to change portion of coarse-grid correction actually imparted to the finest-grid solution. This value was set at 1 throughout the present work.

QBSTEP - (l): If true, a backstep model test case is simulated; if false, another case is simulated. This test case is just in the formulation stages to date.

QBUMP - (l): If true, a half-cylinder model test case is simulated. This test case is untested to date.

QCAV - (1): If true, a driven cavity model test case is simulated. This test case is untested to date.

QCORST - (1): If true, the given model test case is initialized through simulation of 50 iterations on a given coarse grid prior to any other calculation. This option was of little benefit for the cases simulated; however, it was very useful in code debugging.

QCOU - (1): If true, a Couette flow model test case is simulated.

QFREZ - (1): If true, the source term on the right-hand side of each of the momentum equation involving the divergence of the flux field is considered to be zero. This was used in initial code debugging and for the potential flow calculations presented in Chapter 7 only.

QJUNK - (1): If true, a series of diagnostics from numerous subroutines within the code is output in hardcopy form. This was used for debugging only.

QMID - (1): If true, grid stretching across the horizontal axis was accomplished, starting at the middle of the grid and moving outward in both directions.

QMULT - (1): If true, the multigrid technique was employed for the given simulation.

QNADV - (1): If true, all of the advective terms in the equations of motion were considered to be negligible. This was used only for the simulation of potential flow in Chapter 7.

QPRINT - (1): If false, the printing of intermediate solution information every NPREIT iterations was suppressed regardless of the value of NPREIT.

QSLIP - (l): If true, the designation of some portions of no-slip boundaries as slip is allowed; if false, all no-slip boundaries are fully no-slip.

QVENT - (l): If true, a converging-diverging channel model test case was simulated. This scenario has had no evaluation to date and is extemely preliminary.

QZERO - (l): If true, the initial values of pressure and velocity in the field were set to zero.

RHOZERO - (f): Initial density of water in units compatible with the initial velocity specification (see UINIT).

SFLUX - (f): Multiplier for the U flux on the inlet and outlet boundaries; its value was either 1 or 0, signifying either a flux or zero-flux boundary, respectively.

TFLUX - (f): Multiplier for the V flux on the top and bottom boundaries; its value was either 1 or 0, signifying either a flux or zero-flux boundary, respectively.

UINIT - (f): Initial value of the u velocity on the inlet boundary in either English or SI units.

VINIT - (f): Initial value of the v velocity on the inlet boundary in the same units as UINIT.

XANG - (f): Angle, in radians, of skewness of a given grid. The angle is measured from the vertical axis, moving clockwise.

XBOXLEN - (f): Physical length, in the same length units as those of UINIT, of the domain being simulated, in the logical x-direction.

XBUMP1 - (f): X physical location of the upstream-most side of a half-cylinder. The units of this length are those of XBOXLEN.

XBUMP2 - (f): X physical location of the downstream-most side of a half-cylinder. The units of this length are those of XBOXLEN.

XNU - (f): Value of kinematic viscosity of water in units consistent with the designated velocity and characteristic length.

YBOXLEN - (f): Physical length, in the same length units as of XBOXLEN, of the domain being simulated, in the logical y-direction.

YBUMPMX - (f): Physical height, in the same length units as XBOXLEN, of a half-cylinder or physical protrusion.

A.5 OUTPUT SPECIFICATION

Several types of output are possible from this code. Within the subroutine OUTPUT, basically two types are obtained: data files and hardcopy information. Four of the five data file types are discussed above in the input section. The fifth output data file, which always begins with the letters "JN", is a junk file housing the locations of the norms for every iteration. These five data files are extremely important, for they house all the information needed for visualization of the study results.

The hardcopy information presented by subroutine OUTPUT includes a listing of the final velocity, pressure, and streamfunction field values. Also listed are the final U and V flux values. Additionally, several run diagnostics and descriptors are presented such as: number of node points in each dimension; x and y spatial step sizes; number of grids utilized; number of relaxation sweeps on the finest and coarse grids; convergence tolerance; time step multiplier; total number of iterations accomplished; central processing times required to reach the convergence tolerance for each norm; the total central processing time required to run the given number of iterations; the

pseudocompressibility coefficient multiplier; the norms for each flux, pressure, and the divergence of the velocity field at the final iteration; the maximum streamfunction value, and its location. Additional details, such as the test case being simulated, and the Reynolds number and kinematic viscosity for the given run, are also presented.

As a final check, at the end of every printed and stored output data file, an additional namelist named DATAOT is printed. This namelist houses approximately the same information as DATAIN above. It is used as a check in the plotting routines to insure that proper files are being input for the required graphical presentation.

With regard to graphical output, all plotting was done on the VAX 11/750 at the Waterways Experiment Station's Hydraulics Laboratory. Plotting routines using the specific versions of DISPLAA and GCS, two plotting routines available on the VAX, were used to generate plots of convergence histories, vectors, contours, and grids. These routines are somewhat specific to the VAX and lack the general portability to support their presentation herein.

APPENDIX B

DETAILS OF VISCOUS COMPUTATIONS

B.1 INTRODUCTION

This appendix gives details of the numerical formulations for three viscous-flow components: (a) the Laplacian formulation for the viscous terms in generalized curvilinear coordinates assuming slip conditions; (b) the additional viscous contribution to (a) resulting from the effects of no-slip boundaries, and its restriction within the multigrid methodology; and, (c) the formulation of the velocity boundary condition resulting from the no-slip boundary interaction.

B.2 GENERALIZED LAPLACIAN FORMULATION

The viscous terms for the Navier-Stokes equations can be represented mathematically as the product of viscosity and the Laplacian of the velocity field. This Laplacian can be expressed in generalized curvilinear coordinates by recalling Equation 4.19

$$\frac{\nabla \cdot (\nabla u)}{J} = \frac{\nabla^2 u}{J} = \frac{\partial}{\partial \xi} [y_\eta u_x - x_\eta u_y] + \frac{\partial}{\partial \eta} [x_\xi u_y - y_\xi u_x] \qquad (B.1)$$

where each of the terms are as explained in Chapter 4. The cartesian derivatives shown are expressed in generalized coordinates through Equations 4.13 and 4.14 such that the above expression becomes

$$\frac{\nabla^2 u}{J} = \frac{\partial}{\partial \xi} [y_\eta (\xi_x u_\xi + \eta_x u_\eta) - x_\eta (\xi_y u_\xi + \eta_y u_\eta)] + \qquad (B.2)$$

$$\frac{\partial}{\partial \eta} [x_\xi (\xi_y u_\xi + \eta_y u_\eta) - y_\xi (\xi_x u_\xi + \eta_x u_\eta)]$$

Equation B.2 can be better grasped by evaluating the gradient operator (∇) in the Laplacian via the Chain Rule (Equations 4.13 and 4.14), and the divergence operator ($\nabla\cdot$) via the Gauss Divergence Theorem. This results in an alternate expression for Equation B.1 of

$$\frac{\nabla \cdot (\nabla u)}{J} = E_\xi + F_\eta \tag{B.3}$$

where

$$E = J[y_\eta(y_\eta u_\xi - y_\xi u_\eta) - x_\eta(x_\xi u_\eta - x_\eta u_\xi)] \tag{B.4}$$

$$F = J[x_\xi (x_\xi u_\eta - x_\eta u_\xi) - y_\xi(y_\eta u_\xi - y_\xi u_\eta)] \tag{B.5}$$

Again using the north, south, east, and west (n, s, e, w) notation discussed in Chapter 5, Equation B.3 can be expressed as

$$\frac{\nabla \cdot (\nabla u)}{J} = E_e - E_w + F_n - F_s \tag{B.6}$$

The terms E and F, which are proportional to normal derivatives of the respective velocities (u and v) on the cell faces, represent components of momentum flux due to shear stress. Re-introducing the double subscript notation (cc,ee,ww,nn,ss,ne,nw,se,sw) of Chapter 6 to denote coefficients and values of cell-centered quantities in cells adjacent to the cell undergoing computations (cc), the following expressions are presented:

$$u^e_\xi = u_{ee} - u_{cc} \tag{B.7}$$

$$u^n_\eta = u_{nn} - u_{cc} \tag{B.8}$$

$$u^e_\eta = 0.25\ (u_{ne} - u_{se} + u_{nn} - u_{ss}) \tag{B.9}$$

$$u_{\xi}^{n} = 0.25\ (u_{ne} - u_{nw} + u_{ee} - u_{ww}) \tag{B.10}$$

Analogous expressions for the south and west derivatives exist, of course.

Incorporating Equations B.7-B.10 into Equation B.6, and using Equations 4.15-4.18 to re-define the metrics, Equation B.6 takes the form

$$\frac{\nabla \cdot (\nabla u)}{J} = A_{ee}u_{ee} + A_{ww}u_{ww} + A_{ss}u_{ss} + A_{cc}u_{cc} + A_{ne}u_{ne} + A_{nw}u_{nw} + A_{nn}u_{nn} + A_{se}u_{se} + A_{sw}u_{sw} \tag{B.11}$$

The A coefficients are given by

$$A_{cc} = -(a_e + a_w + b_n + b_s) \tag{B.12}$$

$$A_{ee} = a_e - g_n + g_s \tag{B.13}$$

$$A_{ww} = a_w + g_n - g_s \tag{B.14}$$

$$A_{nn} = b_n - g_e + g_w \tag{B.15}$$

$$A_{ss} = b_s + g_e - g_w \tag{B.16}$$

$$A_{ne} = -g_e - g_n \tag{B.17}$$

$$A_{nw} = g_w + g_n \tag{B.18}$$

$$A_{se} = g_e + g_s \tag{B.19}$$

$$A_{sw} = -g_w - g_s \tag{B.20}$$

where

$$a = J\ (x_{\eta}^{2} + y_{\eta}^{2}) \tag{B.21}$$

$$b = J\ (x_{\xi}^{2} + y_{\xi}^{2}) \tag{B.22}$$

$$g = 0.25\ J\ (x_{\xi}x_{\eta} + y_{\xi}y_{\eta}) \tag{B.23}$$

Each of these coefficient expressions are valid as shown only for cells not adjacent to boundaries. Bernard (1988) has shown that, for boundaries which have fluxes which are known at every instant (as is the case for the work in this report), the Laplacian for a cell whose east face lies along a boundary would be

$$\frac{\nabla \cdot (\nabla u)}{J} = -E_w + F_n - F_s \qquad (B.24)$$

Likewise, if the north and east faces of a cell lie on boundaries, the Laplacian is expressed as

$$\frac{\nabla \cdot (\nabla u)}{J} = -E_w - F_s \qquad (B.25)$$

Analogous expressions exist for cells whose south or west faces lie coincident with known-flux boundaries. The jest of these expressions is, then, that, for boundaries with known fluxes, the lack of a required flux correction there dictates, in part, that there be no momentum flux due to shear through the cell faces coincident with the boundaries. Thus, the individual E or F expressions for the boundary sides are, by definition, zero. The impact of this is the re-definition of the A coefficient matrix above while taking into account the boundary conditions. For example, for the case given in Equation B.25, the A coefficients become

$$A_{cc} = -a_w - b_s + 2g_w + 2g_s \qquad (B.26)$$

$$A_{ee} = 0. \qquad (B.27)$$

$$A_{ww} = a_w + 2g_n - 2g_s \qquad (B.28)$$

$$A_{nn} = 0. \qquad (B.29)$$

$$A_{ss} = b_s + 2g_e - 2g_w \tag{B.30}$$

$$A_{ne} = 0. \tag{B.31}$$

$$A_{nw} = 0. \tag{B.32}$$

$$A_{se} = 0. \tag{B.33}$$

$$A_{sw} = -2g_w - 2g_s \tag{B.34}$$

B.3 CALCULATION AND RESTRICTION OF NO-SLIP CONTRIBUTION

For cells adjacent to stationary, no-slip boundaries, the normal component of momentum flux through the boundary is computed subject to the constraint that both velocity components are zero on the boundary. This factor is taken directly into account in the formulation of the velocity boundary condition presented in the next section. A special modification of this formulation for the case of of a moving top boundary (as in the case of the Couette flow simulations) is also presented. The consequence of these formulations is that velocity values are assigned in fictitious cells just beyond the actual physical boundaries for computational purposes. Since the velocity is uniformly zero (or constant for the Couette case) for all cell faces tangent to the boundary, the ξ-derivatives for a north or south cell face coincident with a boundary would be zero. Likewise, the η-derivatives for an east or west cell face tangent to a boundary would be zero. This greatly simplifies the shear-stress momentum flux through the given cell face coincident to the boundary. Thus, for a cell whose south face is tangent to a boundary, the component of F_s in the Laplacian reduces to

$$F_s = b_s(u(i,j) - u(i-1,j)) \tag{B.35}$$

where the u(i-1,j) value is ficticious u-velocity computed as shown in

the next section, and all other terms are as defined above. Similar expressions are computed for each cell face tangent to a no-slip boundary. The result of this computation is an additional term which is added to the Laplacian computed in the previous section. This term accounts for the shear force component generated on no-slip surfaces which is in addition to the shear calculated in the previous section.

In order to simulate this additional shear force on coarse grids within the multigrid cycle, these additional terms are computed on the finest grid for every cell face coincident to a no-slip boundary. These values are then integrated, just as the fluxes are as discussed in Chapter 5 of the main text, along the number of finest-grid boundary cell faces corresponding to a single coarse-grid boundary cell face. This integration is done from the finest grid to each coarse grid. In this way, the shear force generated on the finest grid by the no-slip boundary/fluid interaction is transmitted to each coarse grid.

B.4 VELOCITY BOUNDARY CONDITION FOR NO-SLIP BOUNDARIES

Investigation of the Couette flow in the presence of a pressure gradient case study, discussed in Chapter 8, revealed the need to utilize a second-order velocity boundary condition rather than a first-order one. The need for the second-order condition was signaled by the incorrect computation of the shear stress, when compared to the analytical solution, when using first-order approximations. The second-order approximation was computed for the possibilities of east, west, north, or south cell faces being coincident with no-slip boundaries. In addition, the possibility of a moving top boundary, with an initial u velocity of u_0 and an initial v velocity of 0, was incorporated into the boundary condition formulation. As each of these formulations

was done in the same manner, the development of the boundary approximation for a north cell face residing on the top no-slip boundary is given. Included is the possibility of a moving top boundary.

The formulation is begun by expressing the velocity at a given location (i,j+1) as a function of other known values through the Taylor Series expansion below

$$u(i,j+1) = u(i,j) + u_y(i,j)\,\Delta y + u_{yy}\,\frac{\Delta y^2}{2} \qquad \text{(B.36)}$$

where each of the derivatives is evaluated at the point (i,j). Recall that the top boundary, which is formally at (i,j+1/2), has a known constant u velocity (u_0) that is not necessarily zero. From this, and the equation above, the Taylor series expression for the velocity on the top boundary in terms of the fictitious velocity "beyond" the boundary u(i,j+1) is

$$\begin{aligned} u(i,j+\tfrac{1}{2}) &= u_0 \\ &= u(i,j+1) - u_y\,\frac{\Delta y}{2} + u_{yy}\,\frac{\Delta y^2}{8} \end{aligned} \qquad \text{(B.37)}$$

where each of the derivatives is evaluated at (i,j+1). These derivatives are formulated as standard second-order differences, the first derivative being, of course, one-sided:

$$u_y = \frac{1}{2\Delta y}\,(3u(i,j+1) - 4u(i,j) + u(i,j-1)) \qquad \text{(B.38)}$$

$$u_{yy} = \frac{1}{\Delta y^2}\,(u(i,j+1) - 2u(i,j) + u(i,j-1)) \qquad \text{(B.39)}$$

Inserting Equations B.38 and B.39 into B.37, the following is derived:

$$8u_0 = 3u(i,j+1) + 6u(i,j) - u(i,j-1) \qquad \text{(B.40)}$$

Then, expressing this equation in terms of u(i,j+1), which is the unknown fictitious velocity value stored just "beyond" the boundary, the following boundary condition is obtained:

$$u(i,j+1) = \frac{(8u_0 - 6u(i,j) + u(i,j-1))}{3} \qquad (B.41)$$

For cases where the velocity condition is one of zero velocity, this expression simplifies to the boundary condition given in Chapter 6.

APPENDIX C

PRESSURE PROLONGATION PROCEDURE

C.1 INTRODUCTION

The pressure prolongation procedure makes use of bilinear interpolation to transfer pressure changes from one coarser grid to its just-finer counterpart. The formulation of the bilinear interpolation procedure is explained below. This procedure is identical for both boundary (those cells having faces coincident with physical boundaries) and field points. The values of the fictitious coarse boundary pressure changes are obtained through linear extrapolation. These values are then employed as any field value in the formulation below.

C.2 BILINEAR INTERPOLATION SCHEME

The basic form of the bilinear interpolation scheme is given as below with the pressure change on the just-finer grid listed as the dependent variable.

$$\Delta p = a'(\Delta\xi) + b'(\Delta\eta) + c'(\Delta\xi)(\Delta\eta) + d' \qquad (C.1)$$

where Δp is the change in pressure prolonged from the coarser grid to the finer; $\Delta\xi$ is the change in computational space along the ξ coordinate from some reference point to the location on the just-finer grid; $\Delta\eta$ is the analogous change along the η coordinate; and a', b', c', and d' are coefficients to be determined. The Δp value will be added to the existing change in pressure on the just-finer grid to get the total change in pressure at the given point on the just-finer grid. This total change will then be prolonged to an even finer grid, or added

to the existing pressure on the finest grid to form the new pressure field.

Determination of the coefficients listed above begins by observing Figure C.1. Shown in the figure are grid locations on both the coarser

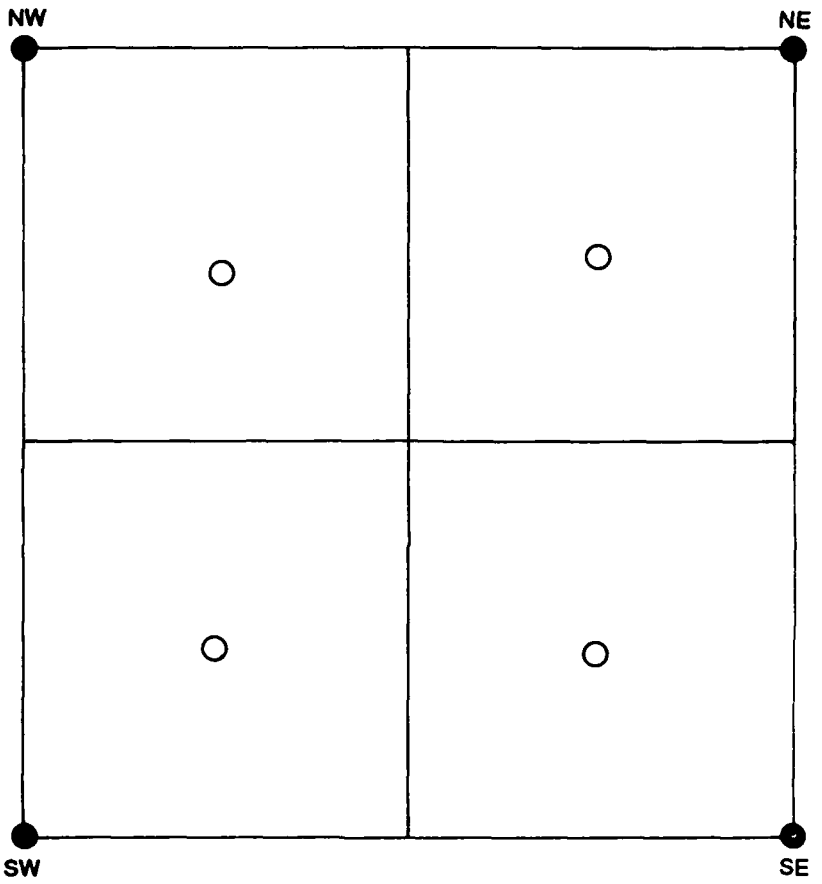


Figure C.1. Fine-to-coarse transfer during prolongation

and finer grids. The locations on the coarser grid are those where the coarser-grid pressure changes are known. The finer-grid locations are those locations to which the coarser changes are to be prolonged. The four coarser-grid locations are designated as southwest (sw), southeast (se), northwest (nw), and northeast (ne). Using the (sw) corner as the

arbitrary reference point, it is clear that the coordinates of each of the coarser-grid points, in terms of $\Delta\xi$ and $\Delta\eta$ units, are as shown below:

$$sw = (0,0) \tag{C.2}$$

$$se = (2,0) \tag{C.3}$$

$$nw = (0,2) \tag{C.4}$$

$$ne = (2,2) \tag{C.5}$$

Further, at each of these locations there is a known coarser-grid pressure change value, designated hereafter as Δp_{sw}, Δp_{se}, Δp_{nw}, Δp_{ne}, respectively. These values are then used to compute the a', b', c', and d' coefficients in Equation C.1 as shown below.

At (0, 0), Δp is equal to Δp_{sw}. Incorporating this into Equation C.1 yields

$$d' = \Delta p_{sw} \tag{C.6}$$

At the southeast corner, Δp is equal to Δp_{se}. Incorporating this and Equation C.6 into C.1 yields

$$a' = \frac{(\Delta p_{se} - \Delta p_{sw})}{2} \tag{C.7}$$

The value of Δp at the northwest corner is Δp_{nw}. Plugging this and Equations C.6 and C.7 into Equation C.1 gives

$$b' = \frac{(\Delta p_{nw} - \Delta p_{sw})}{2} \tag{C.8}$$

Finally, the value of Δp at the northeast corner (2,2) is Δp_{ne}. Plugging this, and Equations C.6-C.8, into C.1 produces

$$c' = \frac{(\Delta p_{ne} + \Delta p_{sw} - \Delta p_{se} - \Delta p_{nw})}{4} \tag{C.9}$$

Using the values for a', b', c', and d' as given above, Equation C.1

becomes

$$\Delta p = 0.5\ (\Delta p_{se} - \Delta p_{sw})(\Delta \xi) + 0.5\ (\Delta p_{nw} - \Delta p_{sw})\ (\Delta \eta) + \qquad (C.10)$$

$$0.25\ (\Delta p_{ne} + \Delta p_{sw} - \Delta p_{se} - \Delta p_{nw})\ (\Delta \xi)(\Delta \eta) + \Delta p_{sw}$$

C.3 LOCATION OF FINER-GRID POINTS

Equation C.10 will provide the mechanism for prolongation only after the location of the finer-grid pressure change locations are known in relation to the reference point in the coarser-grid space: the southwest (sw) corner. Investigation of Figure C.1 shows that, for all finer-grid control volumes not having cell faces that coincide with physical boundaries, the finer-grid pressure change locations are always in the same four locations relative to the southwest corner on the coarser grid. These are given below in ($\Delta \xi$, $\Delta \eta$) coordinate units:

$$\text{pt. } 1 = (0.5,0.5) \qquad (C.11)$$

$$\text{pt. } 2 = (1.5,0.5) \qquad (C.12)$$

$$\text{pt. } 3 = (0.5,1.5) \qquad (C.13)$$

$$\text{pt. } 4 = (1.5,1.5) \qquad (C.14)$$

The boundary-tangent finer-grid cells also have cell centers that have a constant relationship to a coarser-grid southwestern pressure change. For example, the finer-grid cell having west and south faces coincident with physical boundaries has relative coordinates of (1.5,1.5). Similar relations exist for all other boundary-affected finer-grid cells.

APPENDIX D

NOTATION

$\bar{a}$ = vertical distance from top to bottom channel boundaries

a' = constant in bilinear interpolation scheme

a = geometric term in generalized Laplacian

A = matrix of the coefficients of the advective and diffusive terms in the momentum equations

A_{cc} = coefficient matrix for centered term in the generalized Laplacian

b = geometric term in generalized Laplacian

b' = constant in bilinear interpolation scheme

$\bar{c}$ = speed of sound

c = generic constant

c' = constant in bilinear interpolation scheme

d' = constant in bilinear interpolation scheme

DIV = norm of divergence of velocity

DELU = norm of U-flux deviation

DELV = norm of V-flux deviation

E-ab = 10 raised to the power -ab

E = array of viscous components

f = generic scalar function

$\underset{\sim}{f}$ = generic vector function

$\underset{\sim}{f}'$ = forcing terms for a system of differential equations

F = array of viscous components

g = geometric term in the generalized Laplacian

G/F/C = number of grids/number of fine relaxation sweeps/number of coarse relaxation sweeps

i,j = node-point indices

I^{h}_{h+1} = prolongation operator

I^{h+1}_{h}, I^{k+1}_{k} = restriction operators

J = jacobian of coordinate transformation

MOD = mod of a given function

NGRIDS = total number of grids utilized

p = dynamic pressure divided by density

r,s = shift indices

$\underset{\sim}{r}'$ = vector of error residuals

Re = Reynolds number

t = time

u = x-component velocity

$\underset{\sim}{u}$ = vector of cartesian velocities

$\underset{\sim}{u}'$ = intermediate vector of cartesian velocities

$\underset{\sim}{u}'_{new}$ = new estimate for the vector of cartesian velocities after multigrid application

u_0 = velocity of moving top boundary in Couette flow

U,V = volumetric flux components

v = y-component velocity

w = z-component velocity

x,y,z = cartesian coordinates

XPTS = number of grid points along x-coordinate

YPTS = number of grid points along y-coordinate

$\bar{y}$ = distance from bottom boundary to a given point in the flow field

β = pseudo-compressibility coefficient

$\underset{\sim}{\varepsilon}$ = error associated with numerical approximation

ξ,η = curvilinear coordinates

ρ = density of water

ϕ = product of p and the time step

ϕ' = product of Δp and the time step

ν = kinematic viscosity

ω = relaxation coefficient

Superscripts

e,n,s,w = east, north, south, and west faces of a given control volume

h,k = grid level indicators

m = iteration count

n = time level indicator for a given flow variable

* = intermediate predicted value

,* = intermediate time-level values in predictor-corrector sequence

Subscripts

nn,ss,ee,ww,se, ne,sw,nw,cc = cell-center locations for control volumes adjacent to cell (cc) corresponding to north, south, east, west southeast, northeast, southwest, north, and center, respectively

$( )_t$ = partial derivative with respect to t

$( )_x$ = partial derivative with respect to x

$( )_y$ = partial derivative with respect to y

$( )_{yy}$ = second partial derivative with respect to y

$( )_z$ = partial derivative with respect to z

$( )_\xi$ = partial derivative with respect to ξ

$( )_\eta$ = partial derivative with respect to η

Operators

$\frac{\partial}{\partial \xi}$ = partial derivative with respect to ξ

$\frac{\partial}{\partial \eta}$ = partial derivative with respect to η

D_{ξ} = differential operator with respect to ξ

D_{η} = differential operator with respect to η

L = nonlinear differential operator

L' = discrete numerical approximation to L

Δ = change in a given variable spatially or temporally

∇ = gradient operator

$\nabla \cdot \underset{\sim}{u}$ = divergence of $\underset{\sim}{u}$

$\cdot$ = dot product operator

∇^2 = Laplacian operator

$||$ = absolute value operator

END